



Classification de situations de conduite et détection des événements critiques d'un deux roues motorisé

Ferhat Attal

► To cite this version:

Ferhat Attal. Classification de situations de conduite et détection des événements critiques d'un deux roues motorisé. Mathématiques générales [math.GM]. Université Paris-Est, 2015. Français. NNT : 2015PESC1003 . tel-01271020

HAL Id: tel-01271020

<https://theses.hal.science/tel-01271020>

Submitted on 8 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ PARIS-EST

École doctorale Mathématiques et Sciences et Technologies de
l'Information et de la Communication (MSTIC)

THÈSE

présentée en vue de l'obtention du Grade de

Doctorat de l'Université Paris-Est

Spécialité: Mathématiques

par

Ferhat ATTAL

CLASSIFICATION DE SITUATIONS DE CONDUITE ET DéTECTION DES ÉVÈNEMENTS CRITIQUES D'UN DEUX ROUES MOTORISÉ

Soutenue publiquement le 06 juillet 2015 devant le jury composé de :

M. Olivier COLOT	Professeur (Université Lille 1),	Rapporteur
M. Mustapha LEBBAH	Maitre de conférences-HDR (Université Paris 13),	Rapporteur
M. Pascal LARZABAL	Professeur (Université Paris-Saclay),	Examineur
M. Jean-Luc ZARADER	Professeur (Université Paris 6),	Examineur
M. Stéphane ESPIÉ	Directeur de Recherche (IFSTTAR),	Examineur
M. Allou SAMÉ	Chargé de Recherche-HDR (IFSTTAR),	Examineur
M ^{me} . Latifa OUKHELLOU	Directrice de Recherche (IFSTTAR),	Directrice
M. Abderrahmane BOUBEZOUL	Chargé de Recherche (IFSTTAR),	Encadrant

*À mes parents,
À ma femme,
À la mémoire de mon oncle,*

Remerciement

Je tiens tout d'abord à exprimer mes plus profonds remerciements à ma directrice de thèse M^{me} Latifa OUKHELLOU pour avoir accepté de diriger cette thèse. Je lui exprime toute mon estime et la remercie pour ses précieux conseils et encouragements. J'aimerais également remercier mon encadrant M. Abderrahmane BOUBEZOUL de m'avoir offert un encadrement certes exigeant, mais doublé d'une grande disponibilité.

Je tiens également à remercier les membres du jury : M. Pascal LARZABAL professeur à l'université Paris-Saclay, M. Jean-Luc ZARADER professeur à l'université Paris 6, M. Stéphane ESPIÉ directeur de recherche à l'IFSTTAR et M. Allou SAMÉ chargé de recherche-HDR à l'IFSTTAR ainsi que les deux rapporteurs M. Olivier COLOT professeur à l'université Lille 1 et M. Mustapha LEBBAH maître de conférences-HDR à l'université Paris 13. Merci pour votre lecture attentive de la thèse et vos remarques constructives.

Je tiens aussi à adresser mes remerciements les plus chaleureux à l'ensemble des membres de l'équipe de recherche du LEPSIS, en particulier Abdourahmane, pour son aide et soutien, ses critiques constructives, ses précieux conseils et sa disponibilité à tout moment, Hocine, pour ses précieux conseils, ses critiques, son écoute attentive, sa disponibilité et pour son caractère fait d'un mélange d'humour et de dynamisme, Jyida, pour ses encouragements, ses conseils et pour le temps qu'elle a consacré pour la lecture de mon manuscrit de thèse. J'adresse ma sympathie à l'ensemble des personnes que j'ai pu côtoyer durant cette thèse : Enoch, Céline, Élise, Sophie, Kevin, Ghassan, Laurent, Karine, Vérane, Amit, Nassim, Manuel, Lancelot ...

Je tiens à exprimer mes plus profonds remerciements à mes parents pour le soutien et la confiance qu'ils m'ont accordé tout au long de mes études, ainsi que pour m'avoir donné l'opportunité de choisir et d'orienter ma formation en toute liberté. J'adresse également un immense merci à mes frères et mes sœurs qui m'ont encouragé et soutenu de loin depuis toujours.

Enfin et pas la moindre, je tiens à exprimer mes profonds sentiments et mes remerciements à ma femme pour son soutien et son dévouement inconditionnels.

Ferhat ATTAL
Paris, le 10 juillet 2015

Résumé de thèse

Résumé

L'objectif de cette thèse est de développer des outils d'analyse de données recueillies sur les deux roues motorisés (2RMs). Dans ce cadre, des expérimentations sont menées sur des motos instrumentées dans un contexte de conduite réelle incluant à la fois des conduites normales dites naturelles et des conduites à risques (presque chute et chute). Dans la première partie de la thèse, des méthodes d'apprentissage supervisé ont été utilisées pour la classification de situations de conduite d'un 2RM. Les approches développées dans ce contexte ont montré l'intérêt de prendre en compte l'aspect temporel des données dans la conduite d'un 2RM. A cet effet, nous avons montré l'efficacité des modèles de Markov cachés.

La seconde partie de cette thèse porte sur le développement d'outils de détection et de classification hors ligne des événements critiques de conduite, ainsi que, la détection en ligne des situations de chute d'un 2RM. L'approche proposée pour la détection hors ligne des événements critiques de conduite repose sur l'utilisation d'un modèle de mélange de densités gaussiennes à proportions logistiques. Ce modèle sert à la segmentation non supervisée des séquences de conduite. Des caractéristiques extraites du paramètre du modèle de mélange sont utilisées comme entrées d'un classifieur pour classer les événements critiques. Pour la détection en ligne de chute, une méthode simple de détection séquentielle d'anomalies basée sur la carte de contrôle MCUSUM a été proposée. Les résultats obtenus sur une base de données réelle ont permis de montrer l'efficacité des méthodologies proposées à la fois pour la classification de situations de conduite et à la détection des événements critiques de conduite.

Mots—clés:

Deux roues motorisés (2RMs), classification de situations de conduite, apprentissage supervisé et non—supervisé, détection et classification hors ligne des événements critiques de conduite, détection en ligne des chutes d'un 2RM, détection d'anomalies.

Table des matières

Table des matières	v
Glossaire des notations	vi
Glossaire des abréviations	viii
1 Introduction générale	1
1.1 Contexte général	1
1.2 Organisation de la thèse	2
2 Contexte applicatif et problématique de l'analyse de données d'un 2RM	5
2.1 Introduction	6
2.2 Évolution de l'accidentalité des 2RMs	6
2.2.1 Comparaison deux-roues motorisés (2RM)/véhicules légers (VL)	6
2.2.2 Catégorisation des accidents de deux-roues motorisés	8
2.2.3 Particularités et difficultés de la conduite d'un 2RM	9
2.2.4 Projets de recherche sur la sécurité des 2RMs	10
2.2.5 Méthodologie proposée pour la problématique des 2RMs	12
3 Expérimentation et acquisition de données	13
3.1 Introduction	14
3.2 Description de la moto instrumentée	14
3.2.1 Les différents niveaux d'acquisition	14
3.2.2 Enregistreur de données	17
3.2.3 Logiciel de traitement des données	18
3.3 Données recueillies	18

3.3.1	Séquences de conduite normale	18
3.3.2	Séquences de conduite à risque "presque chute"	19
3.3.3	Séquences de conduite avec chute	21
3.4	Construction de la base de données	23
3.5	Pré-traitement de données	24
3.5.1	Remplacement des valeurs manquantes et aberrantes	24
3.5.2	Filtrage des données	24
3.5.3	Annotation/Normalisation des données	26
3.6	Conclusion	29
4	Éléments sur la théorie de l'apprentissage	33
4.1	Introduction	34
4.2	Généralités sur l'apprentissage	34
4.2.1	Apprentissage supervisé	34
4.2.2	Apprentissage non supervisé	37
4.3	Sélection des caractéristiques	38
4.3.1	Phase de sélection des meilleurs sous ensembles	39
4.3.2	Phase de validation	42
4.4	Évaluation des performances	42
4.4.1	Mesure de la performance de généralisation	42
4.4.2	Mesure de la performance d'un classifieur	44
4.5	Conclusion	48
5	Classification de situations de conduite d'un 2RM	49
5.1	Introduction	50
5.1.1	État de l'art sur la classification de situations de conduite	50
5.1.2	Méthodologie de la classification de situations de conduite d'un 2RM	53
5.2	Modèle de mélange de Gaussiennes	53
5.2.1	Modèle de mélange	53
5.3	Les k -plus proches voisins (k -ppv)	58
5.4	Séparateurs à Vaste Marge	59
5.5	Forêts aléatoires	61

5.6	Approche adaptée aux données séquentielles	63
5.6.1	Généralités sur les Chaînes de Markov	63
5.6.2	Modèle de Markov caché discret	64
5.6.3	Modèle de Markov caché continu	68
5.7	Résultats de la classification de situations de conduite d'un 2RM	69
5.7.1	Critères d'évaluation des performances des classifieurs	69
5.7.2	Apprentissage des modèles	70
5.7.3	Sélection des hyper-paramètres des modèles	70
5.7.4	Évaluation des résultats et discussion	72
5.8	Conclusion	78
6	Détection des évènements critiques dans la conduite d'un 2RM	79
6.1	Introduction	80
6.2	Détection et classification hors ligne des évènements critiques dans la conduite d'un 2RM	80
6.2.1	Formalisation du problème de détection et de classification hors ligne des évènements critiques de conduite	81
6.2.2	Segmentation de signaux multidimensionnels via le modèle RHLF	82
6.2.3	Stratégie de détection et de classification des évènements critiques de conduite	86
6.3	Détection en ligne des chutes des 2RMs	87
6.3.1	État de l'art sur les systèmes de détection de chute des 2RMs . .	88
6.3.2	État de l'art sur la détection d'anomalies	89
6.3.3	Formalisation de la détection de chute à l'aide des cartes de contrôle	90
6.3.4	Description de la carte de contrôle multivariée à somme cumulée	92
6.4	Application de la méthodologie proposée et résultats	93
6.4.1	Résultats de la détection et la classification des évènements cri- tiques de conduite	93
6.4.2	Résultats de la détection de chute via la carte de contrôle MCUSUM	99
6.5	Conclusion	104
7	Conclusion et Perspectives	107

Bibliographie	119
Table des figures	122
Liste des tableaux	124
Liste de publications	125

Glossaire des notations

\triangleq	Égale par définition.
\mathcal{D}^l	Base de données.
a_x	Accélération longitudinale.
a_y	Accélération latérale.
a_z	Accélération verticale.
r_x	Vitesse de roulis.
r_y	Vitesse de tangage.
r_z	Vitesse de lacet.
\mathbf{x}	Vecteur de mesures, <i>i.e.</i> $\mathbf{x} \triangleq [x_1, x_2, \dots, x_d]^T$.
\mathbf{X}	Matrice de l'ensemble des mesures, <i>i.e.</i> $\mathbf{X} \triangleq \{\mathbf{x}_i\}_{i=1, \dots, N}$.
\mathcal{X}	Espace de représentation des objets (des formes).
\mathcal{Y}	Espace des sorties désirées.
\mathbf{y}	Ensemble des étiquettes, <i>i.e.</i> $\mathbf{y} \triangleq \{y_i\}_{i=1, \dots, N}$.
\mathbf{z}	Variables latentes.
\mathcal{S}	Base d'apprentissage.
f_θ	Fonction cible à apprendre.
N	Nombre d'objets dans la base d'apprentissage.
\mathbb{E}	Espérance mathématique.
\mathcal{F}	Ensemble de caractéristiques.
κ_c	Indice de Kappa.
$f(\mathbf{x})$	Fonction de densité de probabilité.
$f(\mathbf{x}_i; \boldsymbol{\theta})$	Fonction de densité de probabilité paramétrée.
\mathcal{N}	Loi normale.
\mathcal{L}	fonction de log-vraisemblance.
\mathcal{L}_c	fonction de log-vraisemblance complétée.
\mathcal{M}	Loi Multinomiale.
$\boldsymbol{\mu}$	Vecteur des moyennes.
$\boldsymbol{\Sigma}$	Matrice de variance-covariance.

Glossaire des abréviations

ABS	Anti Blocage Système.
ACEM	Association de Constructeurs Européens de Motocycles.
ACP	Analyse en Composantes Principales.
ALD	Analyse Linéaire Discriminante.
AR	Arrêt.
ARL	Average Run Length.
AUC	Area Under Curve.
BAAC	Bulletin d'Analyse des Accidents Corporels.
CAH	Classification Ascendante Hiérarchique.
CART	Classification And Regression Trees.
CBS	Coupled Braking System.
CHMMs	Continuous Hidden Markov Models.
DAMOTO	Advanced Telematics for enhancing the safety and comfort of motorcycle riders.
DHMMs	Discrete Hidden Markov Models.
EDA	Études Détaillées d'Accident.
EM	Expectation-Maximization.
GMMs	Gaussian Mixture Model.
GPS	Global Positioning System.
HMMs	Hidden Markov Models.
IFSTTAR	Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux.
IOHMMs	Input-Output Hidden Markov Models.
IRLS	Iteratively Reweighted Least Squares.
k -ppv	k -plus proches voisins.
LD	Ligne Droite.
LLE	Locally Linear Embedding.
LOOS	Leave One-Out Sequence.
MAP	Maximum A Posteriori.
MCUSUM	Multivariate CUMulative SUM control chart.
MDS	Multi-Dimensional Scaling.

MSF	Motorcycle Naturalistic Study.
MSP	Maitrise Statistique des Processus.
NDS	Naturalistic Driving Studies.
ONISR	Observatoire National Interministériel de la Sécurité Routière.
RAP	Reconnaissance Automatique de la Parole.
RF	Random Forest.
RHLP	Regression model with a Hidden Logistic Process.
ROC	Receiver Operating Characteristics.
RP	Rond-Point.
RSB	Rapport Signal sur Bruit.
SBS	Sequential Backward Selection.
SFS	Sequential Forward Selection.
SIM2CO	Conception de modules de formation aux habiletés c ognitives de c onduite moto sur s imulateur
SPC	Statistical Process Control.
SUMOTORI	Sûreté sécurité de l'ensemble moto/motard contre les risques liés à son environnement.
SVM	Séparateurs à Vaste Marge.
TFDT	Transformée de Fourier Discrète Fenêtrée.
VA	Variable Aléatoire.
VD	Virage à Droite.
VG	Virage à Gauche.
V-I-C	Véhicule-Infrastructure-Conducteur.
VL	Véhicule légers.
2-BE-SAFE	2-wheeler BEhavior and SAFETy.
2RMs	Deux Roues Motorisés.

Chapitre 1

Introduction générale

1.1 Contexte général

Les conducteurs de 2RMs sont parmi les plus vulnérables des usagers de la route. Ils contribuent à hauteur de 24% à la mortalité routière en France pour seulement 2% du trafic routier. Le risque pour un conducteur de 2RM d'être tué sur la route est 20 fois plus important qu'un automobiliste.

Malgré ce sur-risque, le marché du 2RM a fortement augmenté ces dernières années. Le choix d'utiliser un 2RM pour les trajets domicile-travail est essentiellement motivé par, la réduction du temps de trajet et la mise en œuvre de nouvelles politiques de déplacements urbains durables qui visent à limiter l'usage de la voiture individuelle dans les zones urbaines en particulier les centres villes.

Dans ce contexte d'augmentation du trafic des 2RMs, il est nécessaire et indispensable de garantir et d'accroître leur sécurité dans les déplacements. D'autant plus que les conducteurs de 2RM sont considérés comme des usagers vulnérables de par leur manque de protection et de visibilité. Depuis des années, plusieurs études ont permis de mettre en évidence un certain nombre de facteurs de risque liés à ce moyen de transport : vitesse excessive, manque de visibilité de la part des automobilistes, non respect des règles de sécurité, alcool, Tous ces facteurs sont regroupés autour d'un acteur principal du système de conduite : **le conducteur**. Par contre, il subsiste peu d'études sur les comportements de conduite des conducteurs de 2RMs en situation réelle. Ce manque de connaissance sur les pratiques réelles des conducteurs de 2RMs s'explique par le manque d'outils valides et spécifiques pour l'étude des comportements de cette population d'usagers de la route.

De nombreux travaux et projets de recherche sont menés pour aider à une meilleure compréhension des comportements de conduite de 2RM. Ces projets sont basés sur l'étude des comportements de conduite en situations réelles, incluant à la fois des situations de conduite normale (Naturalistic Riding Studies en anglais) ou à risque (chutes

ou presque chutes). Dans le cadre de ces études, des expérimentations sont menées à l'aide de motos instrumentées, afin d'analyser la conduite *a posteriori* en utilisant les signaux collectés.

L'objectif de ces travaux de thèse est de mettre au point des outils d'analyse automatique des signaux (accélérations et vitesses angulaires dans les trois dimensions de l'espace) en se basant sur l'apprentissage statistique.

1.2 Organisation de la thèse

Ce manuscrit débute par le chapitre 2 qui décrit le contexte applicatif et la problématique de l'analyse de données des 2RMs. Dans un premier temps, nous aborderons les enjeux de l'accidentologie des 2RMs ainsi que les particularités et les difficultés liées à la conduite de ce type de véhicule. Ensuite, nous présenterons un état de l'art des travaux de recherche liés à la conduite des 2RMs. Enfin, sur la base de ces éléments, nous décrirons une démarche fondée sur l'exploitation des données de mesures issues d'expérimentations réelles via les capteurs installés sur la moto afin d'identifier des signatures type de conduite (conduite "normale" et conduite "critique").

Le chapitre 3 présente les moyens expérimentaux mis en œuvre dans le cadre de nos travaux afin d'étudier le comportement des conducteurs de 2RMs. Après avoir décrit les différents capteurs installés sur le 2RM et la base de données recueillie, la suite du chapitre introduit les pré-traitements des données qui consistent au remplacement des valeurs manquantes et aberrantes, au filtrage de bruit ainsi que la normalisation des données de mesures. Ces pré-traitements spécifiques sont effectués pour augmenter le pouvoir descriptif et discriminant du phénomène étudié.

Le chapitre 4 est consacré à la description des notions de base de l'apprentissage automatique. Dans un premier temps, nous décrirons l'apprentissage supervisé, tout en détaillant les deux approches discriminative et générative, ainsi que les deux types de problèmes traités par ce type d'apprentissage à savoir: la classification et la régression. Après cela, un autre type d'apprentissage sera décrit: l'apprentissage non supervisé. Nous allons aussi présenter quelques approches de base de sélection des caractéristiques à savoir, l'approche filtre, l'approche *wrapper* et l'approche hybride. Enfin, nous présenterons un ensemble de critères pour l'évaluation des performances des classifieurs.

Le chapitre 5 concerne la problématique de reconnaissance automatique de situations de conduite à l'aide d'approches de classification. La première partie du chapitre est consacrée à la description des approches d'apprentissage utilisées à savoir: le modèle de mélange de gaussiennes, les k -plus proches voisins, les séparateurs à vaste marge, les forêts aléatoires et les modèles de Markov cachés. La suite du chapitre s'attache à présenter la mise en œuvre de ces différentes approches pour la reconnaissance de

quatre situations de conduite: la ligne droite, le virage à gauche, le virage à droite et le rond-point.

La détection d'évènements de conduite à risque fait l'objet du chapitre 6. La première partie est consacrée au problème de détection et de classification hors-ligne des évènements critiques de conduite des 2RMs. Nous l'avons formulé comme un problème de détection de changements de statistiques (moyennes et variances) des signaux capteurs (accéléromètres/gyromètres) d'un régime de conduite à un autre. Une attention particulière a été portée sur les méthodes de segmentation de séries temporelles multidimensionnelles. En premier lieu, deux méthodes de segmentation adaptées à notre problème à savoir: Multi Regression model with a Hidden Logistic Process (MRHLP) et Hidden Markov Model (HMM) sont présentées. Ensuite, nous aborderons la méthodologie de classification des évènements de conduite en se fondant sur l'algorithme des k -ppv. La deuxième partie est consacrée au problème de détection en-ligne de chute des 2RMs. Ce dernier est formulé comme un problème de détection séquentielle d'anomalies. Une méthode de détection séquentielle d'anomalies bien adaptée à notre problème qui est la carte de contrôle MCUSUM (Multivariate CUmulative SUM control chart) sera mise en œuvre.

Une conclusion de ce mémoire rappellera les différentes étapes du travail effectué ainsi que les résultats obtenus et les perspectives de recherche découlant de ces travaux de thèse.

Chapitre 2

Contexte applicatif et problématique de l'analyse de données d'un 2RM

Sommaire

2.1	Introduction	6
2.2	Évolution de l'accidentalité des 2RMs	6
2.2.1	Comparaison deux-roues motorisés (2RM)/véhicules légers (VL)	6
2.2.2	Catégorisation des accidents de deux-roues motorisés	8
2.2.3	Particularités et difficultés de la conduite d'un 2RM	9
2.2.4	Projets de recherche sur la sécurité des 2RMs	10
2.2.5	Méthodologie proposée pour la problématique des 2RMs	12

2.1 Introduction

Dans ce chapitre, nous présentons le contexte pratique ayant motivé ces travaux de thèse. Dans un premier temps, nous mettons en évidence les enjeux de l'accidentologie des 2RMs ainsi que les particularités et les difficultés liées à la conduite de ce type de véhicule. A travers ces particularités et difficultés, nous montrerons la difficulté de développer des systèmes de sécurité dédiés aux 2RMs sans comprendre les comportements réels des conducteurs de 2RMs. Ensuite, nous ferons une synthèse des travaux de recherche dans la littérature sur la conduite des 2RMs. Enfin, sur la base de ces éléments, nous proposons une démarche fondée sur l'exploitation des données de mesures issues d'expérimentations sur site réel via les capteurs installés sur la moto afin d'identifier des signatures type de conduite incluant à la fois des situations de conduite "normales" et des situations "accidentogènes".

2.2 Évolution de l'accidentalité des 2RMs

2.2.1 Comparaison deux-roues motorisés (2RM)/véhicules légers (VL)

Rappelons que les pouvoirs publics se sont fortement mobilisés ces dernières années dans l'ensemble des pays industrialisés et particulièrement en France pour réduire le nombre de morts sur la route. Cette mobilisation se traduit par la mise en place de plusieurs mesures de type contrôle sanction non seulement au niveau de la vitesse (radars fixes et mobiles) mais aussi au niveau de l'alcoolémie, voir figure 2.1.

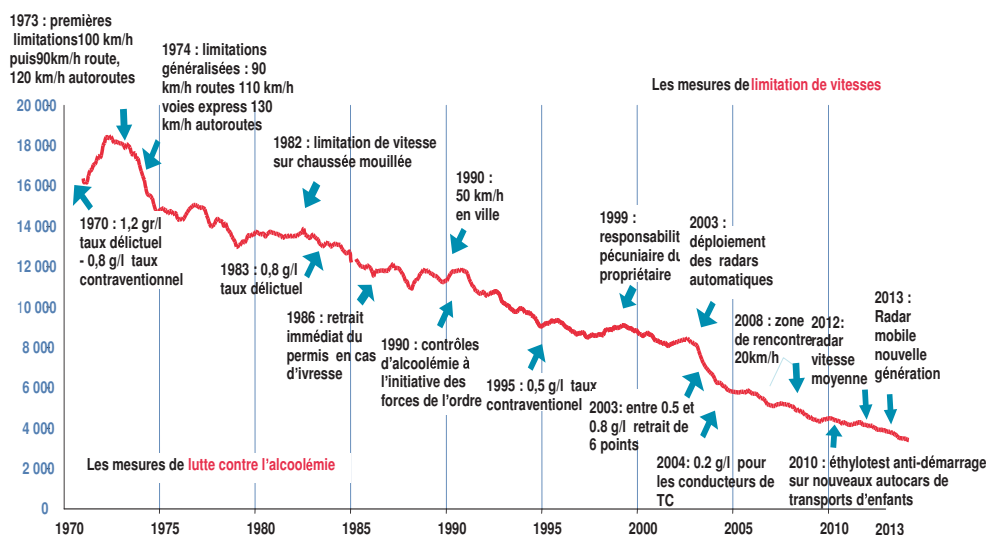


Figure 2.1: Évolution de la mortalité routière en France métropolitaine et les mesures prises en matière de sécurité routière entre 1970 et 2013: source ONISR.

Ces mesures ont permis de réduire considérablement la mortalité routière. Néanmoins,

2.2 Évolution de l'accidentalité des 2RMs

toutes les catégories d'usagers de la route n'ont pas bénéficié de manière homogène de cette tendance baissière de la mortalité routière, voir figure 2.2.

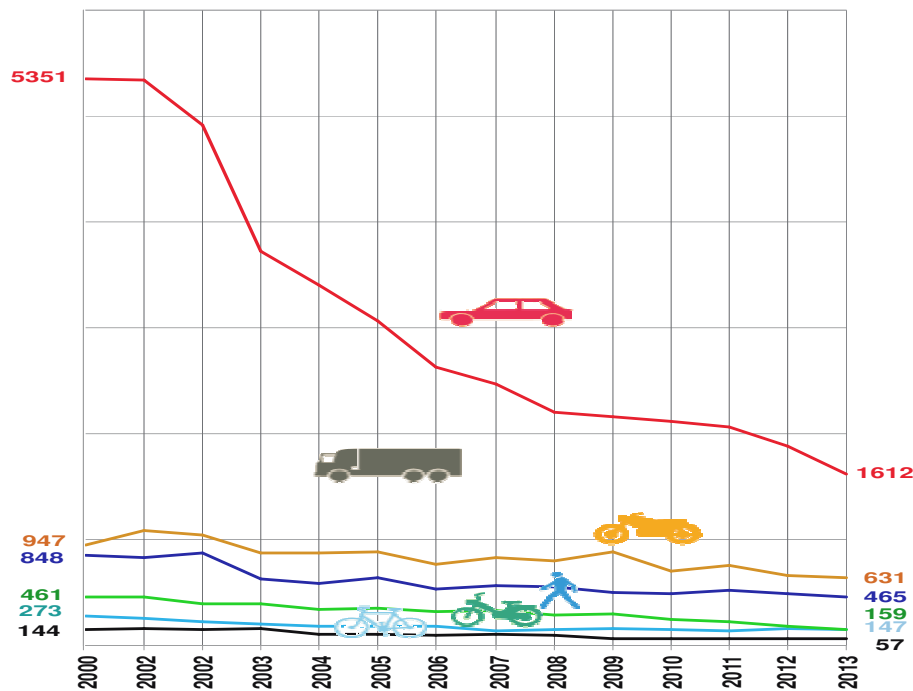


Figure 2.2: Nombre de victimes par catégorie d'usagers de la route: source ONISR.

Notons que pour l'année 2013, les statistiques montrent un bilan alarmant pour les conducteurs de deux-roues motorisés avec près du quart des décès (631 motocyclistes¹ et 159 cyclomoteuristes²) alors qu'ils ne représentent que 2% du trafic routier. Le taux de mortalité des usagers de 2RMs en France est l'un des plus forts taux d'Europe avec 24%. La France se rapproche de l'Italie (26%) malgré un parc de 2RMs nettement moindre (estimé par l'ACEM³ à 3,4 millions en France contre 8,9 millions en Italie).

Les statistiques de la sécurité routière montrent également, que les deux-roues motorisés représentent 31% des blessés hospitalisés dans les fichiers BAAC⁴. En rappelant que les statistiques des fichiers BAAC donnent plus d'informations sur la nature de l'accident, les véhicules impliqués et le contexte routier dans lequel s'est produit l'accident. En plus, cette catégorie d'usagers représente 43% des blessés graves de la route, d'autant plus que ces blessés graves souffrent de blessures avec des séquelles majeures. Notons qu'il existe une réelle surexposition au risque d'être victime d'un accident pour cette population d'usagers de la route par rapport aux autres modes de

¹Conducteurs de deux-roues type scooter dont la cylindrée n'excède pas 125 cm³ et conducteurs de motos dont la cylindrée dépasse 125 cm³.

²Conducteurs de deux-roues motorisés dont la cylindrée ne dépasse pas 50 cm³ et la vitesse maximale par construction 45 km/h.

³Association de Constructeurs Européens de Motocycles (ACEM)

⁴Bulletin d'Analyse des Accidents Corporels

transport.

2.2.2 Catégorisation des accidents de deux-roues motorisés

Selon les statistiques d'accidentologie, nous pouvons catégoriser les accidents impliquant un 2RM en deux grandes familles d'accidents, voir figure 2.3:

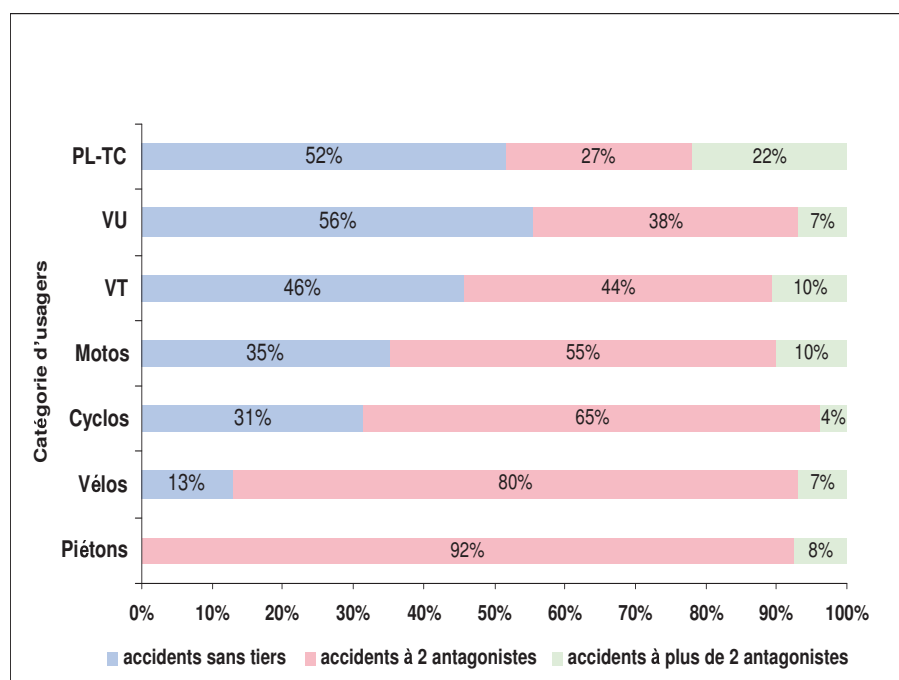


Figure 2.3: Types d'accidents par catégorie d'utilisateurs de la route: source ONISR.

Accidents avec véhicule seul (sans tiers) : Nous entendons par véhicule seul, tout véhicule circulant librement sur la chaussée sans interaction avec un autre véhicule. En terme de nombre de tués sur la route pour l'année 2013, les accidents pour un 2RM seul représentent 35% et 31% des accidents mortels pour les cyclomoteurs et motocyclettes, respectivement. Nous avons exclu de cette catégorie, le cas où un obstacle fixe est heurté. Toutefois, le cas le plus fréquent est un obstacle de type arbre, bâtiment, mur, pile de pont, support de signalisation verticale, poteau, glissières métalliques et barrières en béton, . . .

Accidents de véhicule impliquant d'autres usagers : pour cette catégorie d'accidents, nous constatons que la majorité des accidents mortels des deux-roues motorisés pour l'année 2013 a impliqué un véhicule de tourisme avec 41% pour les motocyclettes et 44% pour cyclomoteurs, voir figure 2.3.

A l'aune de ces statistiques d'accidentologie, nous pouvons noter que les risques liés

à la conduite d'un 2RM constituent un problème majeur de santé publique en France. C'est dans ce sens que la sécurité des usagers de 2RM devient un enjeu pour la société et l'ensemble des acteurs de la sécurité routière.

2.2.3 Particularités et difficultés de la conduite d'un 2RM

Le comportement dynamique spécifique des deux-roues motorisés peut entraîner des particularités et des difficultés dans la conduite vis-à-vis des automobilistes. La spécificité principale de la conduite des 2RMs est la recherche continue de l'équilibre et de la stabilité de la trajectoire. Cette recherche d'équilibre passe par la prise en compte de l'ensemble des interactions du triptyque Véhicule-Infrastructure-Conducteur (V-I-C). La stabilité de la trajectoire est plus complexe pour un 2RM que pour la conduite d'un quatre roues, du fait que le 2RM n'est pas auto-stable par nature. Un 2RM n'est auto-stable que lorsqu'il roule. Cette auto-stabilité s'obtient grâce au phénomène physique appelé «effet gyroscopique» engendré par la rotation d'une roue. Cet effet augmente avec la vitesse de rotation. Plus la vitesse est élevée, plus le 2RM est stable, mais moins il est maniable car son conducteur devra notamment s'opposer à l'effet gyroscopique. En revanche, à faible vitesse l'effet gyroscopique est très faible et par conséquent c'est le conducteur qui assure l'équilibre du 2RM. Cet équilibre est particulièrement précaire surtout lors d'une manœuvre d'évitement ou un freinage brusque ce qui peut entraîner une chute du 2RM [1].

Certains paramètres ont une incidence directe ou indirecte sur l'équilibre et/ou la stabilité du 2RM:

Infrastructure : les pneus assurent la liaison entre le véhicule et la route en transmettant les accélérations, le freinage et le changement de trajectoire. La surface de contact pneu/sol dans les cas des 2RMs est moindre que pour les voitures. Elle correspond approximativement à la taille d'une carte de crédit par pneu. Par conséquent, l'état de la chaussée a un effet direct sur la stabilité du 2RM. Toute dégradation de la chaussée pourra induire une diminution de l'adhérence du pneu et ainsi provoqué la chute du 2RM [1].

Conducteur : dans le cas des 2RMs le conducteur est fortement impliqué dans la tâche de conduite, ce dernier essaye constamment d'assurer la stabilité de son véhicule en essayant de contrer l'effet gyroscopique qui joue un rôle important dans la conduite de moto et en particulier lors de la prise de virage. Pour la prise de virage, nous pouvons noter une grande variabilité dans les pratiques réelles des conducteurs. Toutefois, le comportement produit reste le même sauf que les moyens et les stratégies mis en œuvre par les conducteurs diffèrent d'un conducteur à un autre. La prise d'un virage dans le cas d'un véhicule de type mono-voie, un 2RM par exemple, implique que le conducteur doit incliner son véhicule pour compenser l'inertie engendrée par un changement de

direction. Pour ce faire, il faut donc provoquer un basculement sur l'axe du roulis vers le côté où on veut aller en entreprenant le changement de direction. Ce changement de direction peut être initié, selon les pratiques de chaque conducteur, par un changement de la position du centre de gravité du conducteur, ou bien par un phénomène caractéristique des véhicules deux-roues qui est le contre braquage et qui fait que l'on peut braquer à droite pour incliner le véhicule à gauche et inversement.

En plus de ces particularités et difficultés de la conduite inhérentes à ce type de véhicule, nous avons une autre particularité bien spécifique au 2RM qui est le positionnement multiple sur la chaussée. En effet, le faible gabarit des 2RMs a une incidence sur leur positionnement sur chaussée. Le conducteur d'un 2RM dispose ainsi, plus de degrés de liberté qu'un conducteur de quatre roues dans le positionnement latéral sur la voie de circulation. Le conducteur de 2RM peut positionner son véhicule au milieu de la voie de circulation, par exemple, afin d'avoir une meilleure visibilité et être mieux vu par les autres, ou bien pour éviter les imperfections de la chaussée, ou plus encore afin de pouvoir garder une distance de sécurité latérale.

Le développement d'outils ou de systèmes dans la perspective d'améliorer la sécurité des 2RMs, est une tâche ardue. Cette tâche ne peut se faire sans la phase d'analyse et de compréhension de la conduite (pratiques réelles des conducteurs de 2RMs), car il subsiste de nombreuses inconnues dans les interactions du triptyque Véhicule-Infrastructure-Conducteur (V-I-C). Il existe une nécessité absolue d'approfondir nos connaissances sur les règles comportementales adoptées par les conducteurs de 2RM ou par les autres usagers de la route vis-à-vis de ces véhicules.

2.2.4 Projets de recherche sur la sécurité des 2RMs

Plusieurs études ont permis de mettre en évidence un certain nombre de facteurs de risques liés à ce moyen de transport : vitesse excessive, manque de visibilité par rapport aux autres usagers de la route, non respect des règles de sécurité, alcool, Tous ces facteurs peuvent être liés à un facteur récurrent qui est, le conducteur. La littérature abordant la question des deux-roues motorisés s'est essentiellement développée sur le thème de l'accidentologie ou encore du port du casque et des techniques favorisant ce comportement sécuritaire [2]. En revanche, il existe peu d'études sur les comportements de conduite des conducteurs de 2RM en situation réelle.

Ainsi, de nombreux projets sur l'amélioration de la sécurité des 2RMs ont été réalisés en France et à l'échelle Européenne en se basant sur l'observation des comportements réels. Le projet ANR Predit **SUMOTORI**¹ avait pour objectif, la mise au point de systèmes de sécurité passifs. Dans le projet ANR **DAMOTO**² il s'agissait de définir des critères de déclenchement d'un gilet gonflable. Tandis que le projet européen

¹Sûreté sécurité de l'ensemble moto/motard contre les risques liés à son environnement (2003-2006)

²Détection de situations d'accidents en motocycles (2008-2011)

SAFERIDER ¹ avait pour objectif le développement des systèmes d'assistance à la conduite ou d'aide à la navigation. Le projet européen **2-BE-SAFE**² était orienté vers l'acquisition de connaissances pour une meilleure compréhension des comportements des conducteurs de 2RMs en conduite naturelle avec des véhicules instrumentés.

Dans le cadre des projets cités ci-dessus, des études de conduite naturelle (de l'anglais *Naturalistic Driving Studies (NDS)*) ont été menées. Plusieurs véhicules ont été équipés de capteurs et de systèmes de mesures embarqués de la manière la moins intrusive possible. Ces véhicules ont été prêtés à des volontaires afin de recueillir des données sur leurs trajets habituels (domicile-travail), par exemple. Chaque participant volontaire a conduit son véhicule pendant une longue période. Après une période initiale, le conducteur semble oublier qu'il conduit un véhicule instrumenté et se comporte naturellement, il reprend ces habitudes de conduite.

Plus récemment, d'autres projets nationaux et internationaux qui peuvent s'inscrire dans la même lignée que les projets cités précédemment ont été menés. Le projet ANR SIM2CO+³ qui avait pour objectif, la mise au point des modules de formation aux habiletés cognitives de conduite moto sur simulateur, ainsi qu'un avancement des connaissances sur la cohabitation des 2RMs avec les autres modes de transport routier et sur les processus d'apprentissage à la conduite des motards. Pour cela, les véhicules personnels des motards ont été instrumentés en capteurs et caméras, afin de les suivre pendant deux mois au cours de leurs déplacements habituels. Un programme de recherche américain intitulé (MSF 100 Motorcycle Naturalistic Study [3]) a été lancé. Ce programme de recherche met l'accent sur l'analyse et la compréhension à grande échelle du comportement des conducteurs de 2RMs.

Si les études en conduite naturelle (dite *naturalistic riding*) sont riches d'enseignement en ce qui concerne les comportements réels produits par les conducteurs de 2RMs dans le trafic, toutefois, la quantité de données produite durant ces expérimentations rend ces données mesurées difficilement exploitables de façon manuelle. Le développement d'outils d'analyse de données adaptés à cette problématique est une étape nécessaire pour approfondir nos connaissances sur les comportements de conduite adoptés par les conducteurs de 2RM dans leur conduite et leurs interactions avec les autres usagers de la route. Cette connaissance se révélera primordiale dans tous développements de systèmes dédiés à la sécurité des 2RMs.

¹Advanced Telematics for enhancing the safety and comfort of motorcycle riders (2008-2011)

²2-wheeler BEhavior and SAFETY (2009-2012)

³Conception de modules de formation aux habiletés cognitives de conduite moto sur simulateur (2011-2013).

2.2.5 Méthodologie proposée pour la problématique des 2RMs

À partir de données recueillies en situations de conduite réelles englobant des situations de conduite normales et des situations à risque de type chute ou presque chute, nous visons deux objectifs dans le cadre de cette thèse:

- Le premier est de mettre au point un système de reconnaissance automatique de situations de conduite : ligne droite, virage à gauche, virage à droite et rond-point. En étant capable de faire une telle classification, l'idée est de pouvoir extraire ensuite de façon automatique toutes les situations de type virage à droite ou à gauche par exemple, afin de les analyser plus finement.
- Le deuxième objectif est d'effectuer une détection des événements de conduite à risque de type chute ou presque chute. Cette détection est traitée de deux manières différentes (off-line et on-line), voir figure 2.4.

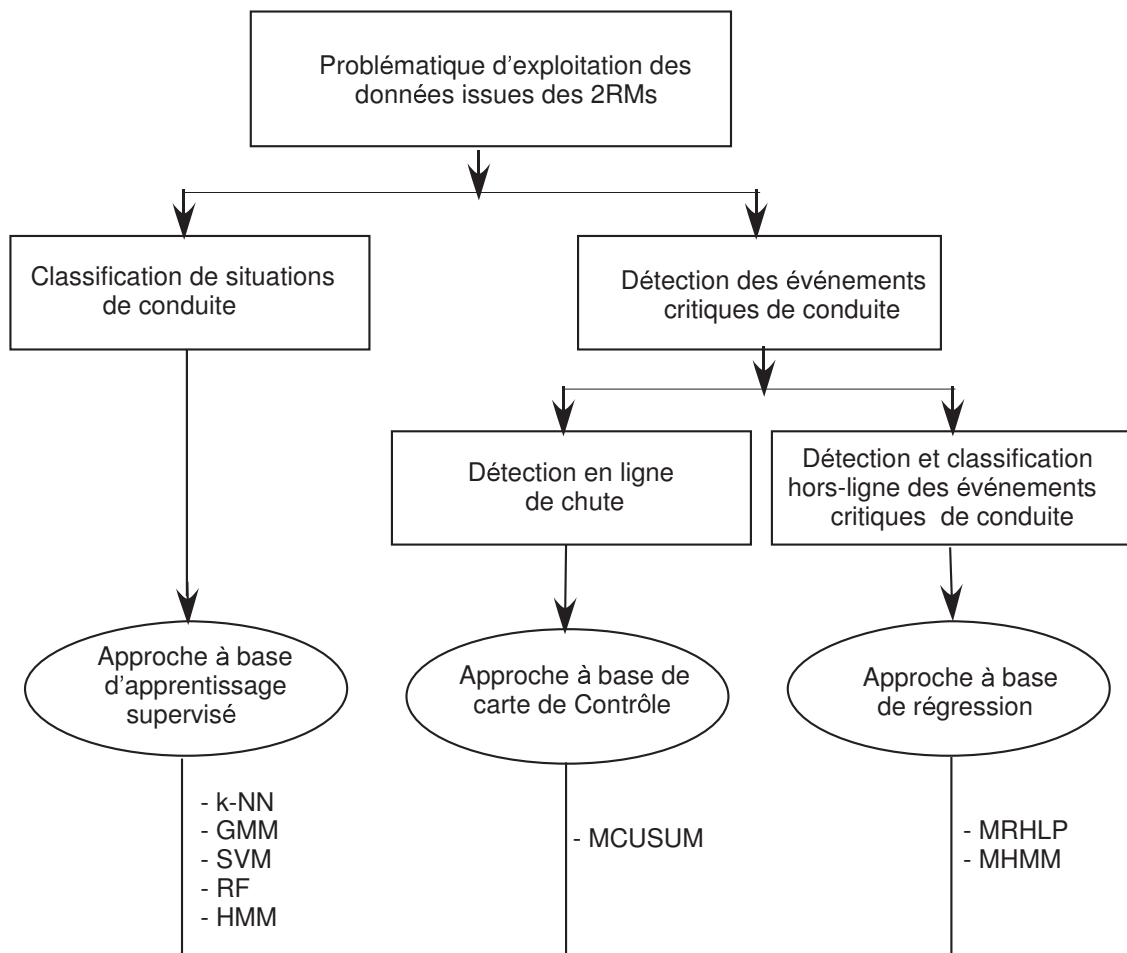


Figure 2.4: Organigramme de l'approche proposée dans ces travaux de thèse.

Chapitre 3

Expérimentation et acquisition de données

Sommaire

3.1	Introduction	14
3.2	Description de la moto instrumentée	14
3.2.1	Les différents niveaux d'acquisition	14
3.2.2	Enregistreur de données	17
3.2.3	Logiciel de traitement des données	18
3.3	Données recueillies	18
3.3.1	Séquences de conduite normale	18
3.3.2	Séquences de conduite à risque "presque chute"	19
3.3.3	Séquences de conduite avec chute	21
3.4	Construction de la base de données	23
3.5	Pré-traitement de données	24
3.5.1	Remplacement des valeurs manquantes et aberrantes	24
3.5.2	Filtrage des données	24
3.5.3	Annotation/Normalisation des données	26
3.6	Conclusion	29

3.1 Introduction

Le but de ce chapitre est de présenter les moyens expérimentaux mis en œuvre dans le cadre de nos travaux de recherche pour étudier le comportement des conducteurs de deux roues-motorisés (2RM).

Le système de conduite constitué par le conducteur, le 2RM, et l'infrastructure où évolue le conducteur peut être caractérisé par une grande quantité d'informations [4]. Le conducteur et son comportement, peuvent être décrits par l'âge du conducteur, son expérience, son état de fatigue et la position de sa tête (direction de son regard) et la force que le conducteur applique sur le guidon et sur les freins (organes de commande). La dynamique du 2RM est décrite par sa position sur la chaussée, sa vitesse et son accélération dans les trois dimensions de l'espace. L'environnement où se déroule la scène de conduite est défini par l'infrastructure, le trafic et les conditions météorologiques. Afin d'avoir un maximum d'informations sur les comportements de conduite réels des conducteurs de 2RM, une instrumentation qui prend en compte les spécificités de ce type de véhicule a été proposée [5]. L'élément central de cette instrumentation est le système d'acquisition de données. Ce système est constitué de capteurs et d'une carte d'acquisition de données. Même si la partie instrumentation a été traitée en dehors de ces travaux de thèse, il nous a semblé intéressant d'en présenter les principales orientations avant d'aborder la partie pré-traitements de données à proprement parler.

3.2 Description de la moto instrumentée

L'établissement IFSTTAR (Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux) dispose des motos instrumentées. Cette instrumentation a été réalisée dans le cadre des projets de recherches ANR SUMOTORI et DAMOTO. Dans la thèse de [5] une instrumentation a été réalisée pour le recueil de données sur la dynamique moto lors de scénarios accidentogènes, afin d'étudier l'accidentologie des 2RM. Pour l'instrumentation de la moto, une architecture capteur a été mise en place afin de mesurer tous les phénomènes de la dynamique moto. Un enregistreur multi-capteurs pour sauvegarder les données issues de ces capteurs a été installé. L'emplacement des capteurs et de l'enregistreur ont été réalisés en vue de rendre l'instrumentation la moins intrusive possible tout en garantissant la pertinence des informations fournies par les capteurs.

3.2.1 Les différents niveaux d'acquisition

Les données utilisées dans cette thèse sont issues d'une moto instrumentée de type HONDA CBF-1000 (voir la figure 3.1) en se basant sur l'architecture définie par [5]. Cette moto est équipée de deux systèmes de sécurité ABS (Anti Blocage Système) et

CBS (Coupled Braking System). Dans l'objectif d'utiliser cette moto instrumentée dans plusieurs projets liés soit à l'étude et l'analyse du comportement des conducteurs de 2RMs au sens cognitif, soit à l'étude de l'accidentologie de ce type de véhicule au sens dynamique moto, cette dernière est équipée de capteurs sur trois niveaux d'enregistrement: au niveau du conducteur, au niveau du véhicule et au niveau de l'environnement. La moto est équipée aussi d'un enregistreur de données permettant de sauvegarder les données acquises par les capteurs, l'instrumentation de la moto sera détaillée ci-après.

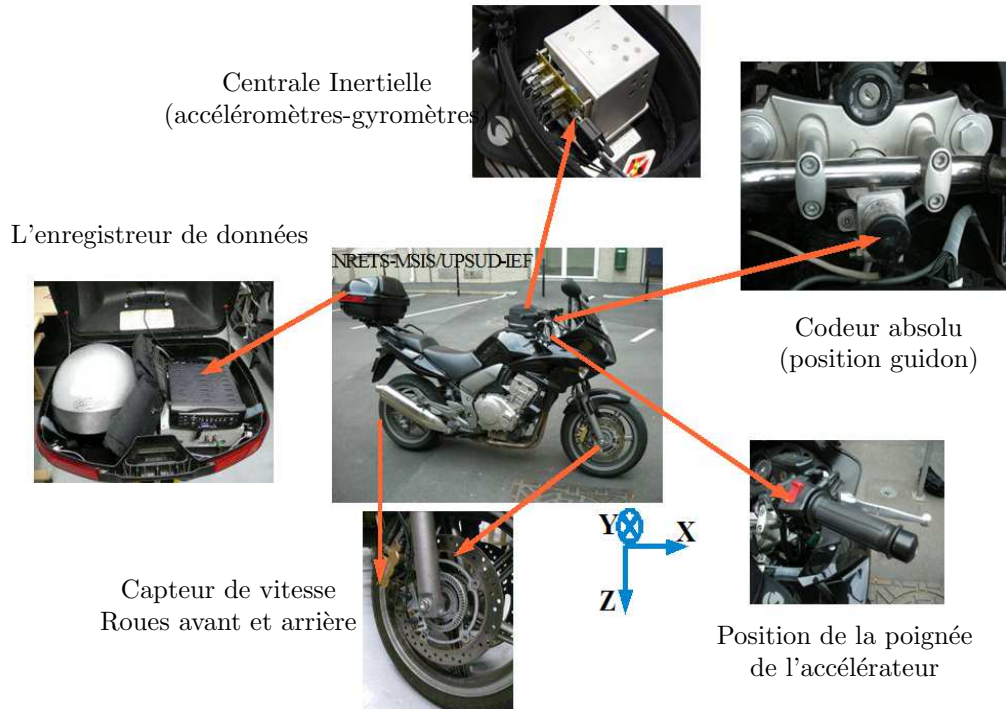


Figure 3.1: Moto instrumentée (Honda CBF1000) et capteurs installés. Les accélérations et les vitesses angulaires sont présentées dans le trièdre de référence.

Acquisition au niveau du conducteur

Ces mesures traduisent les actions entreprises par le conducteur pour agir sur les organes de contrôle et de commande de la moto. Ces actions permettent le plus souvent d'anticiper une situation future [6]. A cet effet, la moto a été équipée de capteurs qui mesurent les actions du conducteur : angle guidon, frein, clignotants, accélérateur.

- **Angle guidon** : l'action du conducteur sur le guidon est mesurée par un codeur absolu (1024) points par tour. Ce codeur a été monté sur la moto par l'intermédiaire d'un pignon placé sur le guidon et une courroie qui le relie au codeur [5]. L'angle guidon est fourni avec une précision de $0,1^\circ$.

- **Frein** : le freinage est l'action du conducteur sur les freins pour ralentir (action sur le frein avant) ou arrêter complètement (action sur le frein avant et arrière) son véhicule. Afin d'éviter les problèmes de certification et d'assurance de la moto, cette information a été récupérée à partir de l'état du feu arrière (éteint/allumé) c'est-à-dire qu'il n'y a pas eu de modifications apportées sur le système de freinage d'origine. Le signal délivré par le feu arrière est donc de type tout ou rien : soit le feu est allumé ce qui implique que le frein est actionné, soit il est éteint et dans ce cas, le frein n'est pas actionné. Cette information nous renseigne juste sur l'état des freins s'ils sont actionnés ou pas.
- **Clignotant** : le clignotant est un dispositif qui permet au conducteur de communiquer ses intentions de conduite aux autres usagers de la route. Son enregistrement présente un intérêt pour l'étude du comportement des conducteurs et la prédiction de ses actions. Comme dans le cas du freinage, cette information est récupérée directement sur l'état (éteint/allumé) des clignotants gauche et droit.
- **Papillon admission air du moteur** : parmi les actions du conducteur mesurées sur la moto, on trouve aussi la puissance du moteur. Cette information peut être récupérée soit à partir de la position de la poignée de l'accélérateur ou bien à partir de la position du papillon admission air du moteur. Pour des raisons de simplicité et de précision le choix a été fait de récupérer la puissance du moteur à partir de la position du papillon admission air du moteur. Cette information est récupérée par l'intermédiaire d'un potentiomètre et d'un montage de mise en forme du signal délivré par ce dernier, voir [5].

Acquisition au niveau du véhicule

Les informations sur la dynamique de la moto traduisent non seulement les conséquences des actions du conducteur sur les organes de contrôle et de commande de la moto mais aussi, son interaction avec l'infrastructure. Certes, il y a une redondance dans ces deux types d'informations (acquisition au niveau du conducteur et acquisition au niveau du véhicule). Mais ces dernières restent plus précises et plus faciles à quantifier que dans le cas des mesures sur les actions de contrôle et de commande. En effet, un changement de vitesse ou d'accélération peut facilement être détecté sur la dynamique de la moto que sur les actions du conducteur du fait de la difficulté de l'analyse de l'accélérateur. Ainsi, la dynamique de la moto peut être mesurée à partir de sa vitesse linéaire et angulaire ainsi que son accélération. Il reste à noter que contrairement aux capteurs au niveau conducteur, les capteurs de la dynamique moto n'engendrent pas de modification des caractéristiques des organes de contrôle commande de la moto.

- **Vitesse longitudinale** : la moto a été équipée d'un capteur de vitesse de type capteur à effet hall qui permet de calculer le nombre de tours de roue (mesures

par pas de 9°). Ainsi, connaissant le diamètre de la roue et le nombre de tours de roue, la distance parcourue et la vitesse peuvent facilement être déduites. Ce capteur a été placé sur les roues avant et arrière de la moto.

- **Accélération et vitesse angulaire :** pour la mesure des accélérations et des vitesses angulaires de la moto, une centrale inertielle a été installée au niveau du réservoir de la moto. Ce système est une combinaison de trois capteurs de type accéléromètre et gyromètre (un capteur accéléromètre qui est intégré dans un capteur gyromètre). Ce capteur fournit à sa sortie un double signal analogique: une accélération avec un étendu de mesure de $\pm 1,8$ g et une vitesse angulaire avec une précision de $100^\circ/s$.

Acquisition au niveau de l'environnement routier

En plus de l'importance des mesures sur les actions du conducteur et la dynamique de la moto, les informations sur l'environnement routier où évolue le conducteur sont aussi nécessaires pour l'analyse et la compréhension de la conduite de moto. Cet environnement est caractérisé par les conditions météorologiques, le type de conduite (diurne/nocturne), le trafic, le contexte routier (ville, campagne, autoroute), l'infrastructure etc. Ainsi, pour prendre en compte le contexte de conduite, la moto a été équipée de caméras et d'un GPS.

- Quatre caméras avec une résolution de 25 images/seconde, ont été installées sur la moto afin de contextualiser l'environnement de la conduite. La scène de devant est filmée par trois caméras sur un champ visuel avant de 160° [7] et la quatrième caméra permet de filmer le visage du conducteur.
- Pour prendre en compte l'infrastructure où se déroulent les expérimentations, la moto a été équipée d'un récepteur GPS de type Garmin qui mesure le positionnement curviligne (la trajectoire) ainsi que la vitesse longitudinale de la moto.

3.2.2 Enregistreur de données

Afin d'enregistrer les mesures issues des capteurs au cours des expérimentations, la moto a été équipée d'un système d'acquisition de données. Ce système est fondé sur l'usage de deux micro-contrôleurs et d'un composant FPGA permettant une acquisition à 1Khz avec un horodatage des données acquises à $4\mu s$. Les données recueillies sont ensuite sauvegardées dans une mémoire flash dont la capacité est d'environ 4 heures.

3.2.3 Logiciel de traitement des données

Dans l'optique d'analyser les données mesurées, un logiciel propriétaire BINASCII a été développé au sein de IFSTTAR. Ce logiciel permet de visualiser et de synchroniser les données des capteurs ainsi que leurs mises en forme (Transformation des données du format binaire au format texte). Par la suite, ces données peuvent être traitées par d'autres logiciels de traitement de signal (Matlab®, ...).

3.3 Données recueillies

Pour mieux analyser l'activité de conduite d'un 2RM, trois types de séquences ont été définis et réalisés sur site réel. Ces trois types de séquences consistent à effectuer des séquences de conduite normale "sans accident", des séquences de conduite de presque chute et des séquences de conduite avec chute.

3.3.1 Séquences de conduite normale

L'objectif de cette campagne de mesures est de collecter des données sur route lors d'une conduite normale (trajets de loisirs, trajets domicile/travail, etc), dans le but de comprendre le comportement des conducteurs face à des situations bien particulières [7]. Afin de se rapprocher le plus possible des conditions de conduite normales, des consignes ont été données aux conducteurs de conduire comme ils ont l'habitude de le faire :

- **Participants :** cinq sujets possédant un permis de conduire moto ont participé à cette expérimentation. Ces participants ont différents profils et expériences de conduite (moyenne d'âge: 39 ans, expérience moyenne de conduite moto: 14 ans).
- **Parcours :** l'acquisition de données a été effectuée dans un milieu urbain (à Paris) voir figure 3.2, sous plusieurs conditions météorologiques (ensoleillé, pluvieux et brumeux). Plusieurs séquences (trajectoires) ont été effectuées par cinq participants (S_1, S_2, S_3, S_4, S_5). Au total, onze séquences notées ($S_{q1}, S_{q2}, \dots, S_{q11}$) ont été réalisées par les participants. Chaque participant a effectué deux séquences, à l'exception, d'un participant qui en a fait trois. Chaque séquence dure sept à huit minutes en moyenne sur une distance totale d'environ 2500 m. La vitesse de la moto est assez variable du fait des nombreuses accélérations et décélérations, voir même des arrêts/redémarrages en raison des feux de circulation et présences des piétons. Les situations de conduite les plus fréquentes rencontrées au cours de ses séquences sont: des arrêts (AR), des lignes droites (LD), des virages à gauches (VG), des virages à droites (VD), des ronds-points (RP).

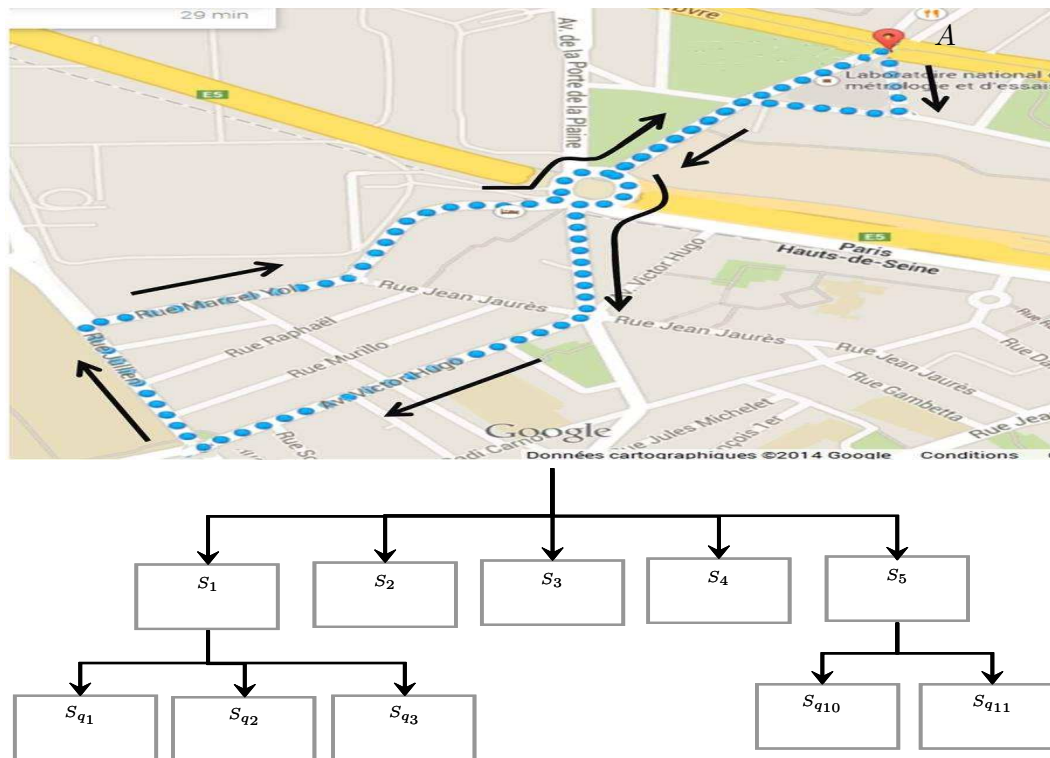


Figure 3.2: Trajet réalisé par les cinq participants en conduite normale.

3.3.2 Séquences de conduite à risque "presque chute"

L'objectif de cette campagne de mesures est d'acquérir des données dans des situations de conduite extrêmes. Ces situations extrêmes se traduisent par une conduite agressive, comme le freinage et l'accélération brusques, les embardées, la perte de contrôle de la moto Cette expérimentation a permis de réaliser 10 séquences sur une distance d'environ 500 m, d'une durée de 2 minutes chacune ont été réalisées par un conducteur expérimenté (instructeur en conduite moto de la police nationale).

- **Participant** : afin de pouvoir mener à bien ces expérimentations et de mieux reproduire les situations de presque chute, un instructeur des motards de la Police Nationale a réalisé les différentes séquences de conduite à risque.
- **Piste** : une partie de ces expérimentations (les scénarios de conduite sportive) se sont déroulées sur le circuit CAROL situé à Tremblay près de Paris (voir figure 3.3) sur une distance de 500 m environ pour une période de 2 minutes pour chaque scénario. Concernant les scénarios de conduite en presque accident, ces essais ont été réalisés par le même cascadeur sur les pistes de la CERAM à Mortefontaine (voir figure 3.4).

Plusieurs scénarios ont été réalisés à savoir:



Figure 3.3: Piste d'essai CAROL, où les expériences de conduite à risque ont été réalisées.

- **Conduite sportive sur piste** : dont l'objectif était de réaliser des tours de pistes en donnant un maximum d'angle d'inclinaison à la moto dans les virages.
- **Conduite en presque accident** dont l'objectif était de:
 1. reproduire les scénarios d'accélération brusque en plein virage,
 2. reproduire les scénarios de freinage brusque dû à la présence/apparition d'un obstacle sur la route,
 3. rouler sur une route fortement dégradée servant à l'évaluation du confort de tenue de route des automobilistes,
 4. reproduire les scénarios de perte de contrôle avec rattrapage de la situation, c'est-à-dire sans que la moto ne tombe,
 5. rouler en zigzag.

Un ensemble de 10 séquences a été réalisé par le conducteur, dans lequel:

- le scénario de conduite sportive a été effectué 3 fois,
- des scénarios d'accélération brusque et l'évaluation du confort de tenue de route ont été effectués 2 fois chacun,
- les scénarios de freinage brusque, de perte de contrôle avec rattrapage de la situation et de la conduite en zigzag ont été effectués une fois.

3.3.3 Séquences de conduite avec chute

Le comportement dynamique d'un 2RM lors d'une chute est complexe et donc très difficile à reproduire par des outils de simulation. Par conséquent, des expérimentations ont été réalisées en:

- reproduisant des chutes à partir de scénarios d'accident,
- mesurant le comportement dynamique du véhicule pendant l'accident.

Scénarios d'accidents

Les chutes sont la conséquence d'une perte de contrôle transversale. Cette perte de contrôle entraîne une chute sur la partie latérale du véhicule. Toute autre chute est la conséquence d'accidents en chocs ou de figures acrobatiques.

L'accidentologie ne détaille pas de manière exhaustive les cas de perte de contrôle amenant à un accident, même au travers de ses EDA (Études Détaillées d'Accident). Par conséquent, le choix des scénarios a été réalisé à travers une étude pragmatique dans le cadre du projet DAMOTO. Cette étude consistait à identifier les cas potentiels de chutes, leurs causes et leurs conséquences. Elle a permis d'établir qu'une chute est caractérisée par les paramètres dynamiques suivants : l'angle d'inclinaison du véhicule, la vitesse d'inclinaison du véhicule.

Ces paramètres caractérisant la dynamique du véhicule ont servi de critères pour sélectionner les scénarios type de perte de contrôle à reproduire expérimentalement.

- **Le cascadeur** : pour rejouer les 5 scénarios de chutes définis dans le paragraphe précédant tout en garantissant la sécurité des intervenants, le bon déroulement des expérimentations et l'intégrité des matériels, un cascadeur professionnel a été engagé. Lors de ces expérimentations un staff médicale était présent sur les lieux d'essais afin de garantir la sécurité et palier aux impondérables.
- **Piste** : les expérimentations se sont déroulées sur les pistes de la CERAM à Mortefontaine (voir figure 3.4) et plus précisément sur l'aire de freinage jouxtant l'anneau de vitesse.

La figure 3.5 illustre un exemple d'essai d'un scénario de chute. La synchronisation des données vidéos et les mesures embarquées a été réalisée au moyen d'un flash fixé sur la moto visible sur le film enregistré par la centrale d'acquisition.

Les 5 scénarios sélectionnés sont présentés ci-après.

1. **Chute à l'arrêt**: la moto est soumise à une chute latérale à une vitesse nulle (vitesse = 0 Km / h). Cette chute se produit généralement au moment de garer la moto ou lorsque le conducteur s'arrête à un feux de circulation.

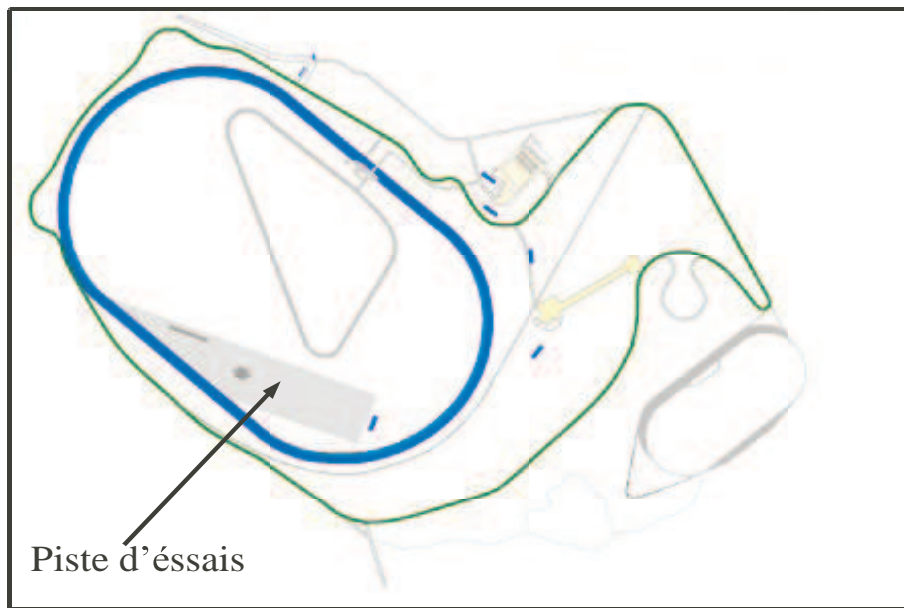


Figure 3.4: Piste d'essai CERAM, où les expérimentations de conduite avec chute ont été réalisées.

2. **Perte d'adhérence en virage:** qui vise à reproduire une chute en plein virage causée par l'action du cascadeur sur le frein avant ce qui provoque le dérapage de la roue avant. Dans ce cas, la vitesse de la moto était de 80 Km/h.
3. **Perte d'adhérence en ligne droite (en présence de la neige ou du verglas):** l'objectif de ce scénario était de reproduire les pertes d'adhérence en ligne droite dues à la neige ou au verglas. Pour ce faire, une instruction a été donnée au cascadeur de tomber en provoquant une chute sur une section de route droite glissante. La chute dans ce scénario est causée par une accélération inappropriée qui provoque le patinage de la roue arrière et par conséquent la perte de contrôle de la moto. Dans ce scénario, la vitesse de la moto était de 30 Km/h.
4. **"Couchage de la moto" (manœuvre intentionnelle):** le scénario ici vise à reproduire l'évitement d'un obstacle inattendu par un conducteur sur ligne droite. Lorsque le freinage n'est pas suffisant pour éviter la collision, le conducteur est obligé de se pencher avec sa moto afin de réduire la gravité de l'accident. Ce scénario a été effectué à une vitesse de 80 km / h.
5. **Perte d'adhérence dans un rond-point:** l'objectif de ce scénario était de reproduire les pertes d'adhérence dans un rond-point qui sont provoquées par une surface glissante telle que les taches d'huile, le gravier, les feuilles d'arbre, la chaussée mouillée, les surfaces en acier (plaques d'égout), Ce scénario se produit généralement dans des rond-points et des intersections. Dans ce cas, une

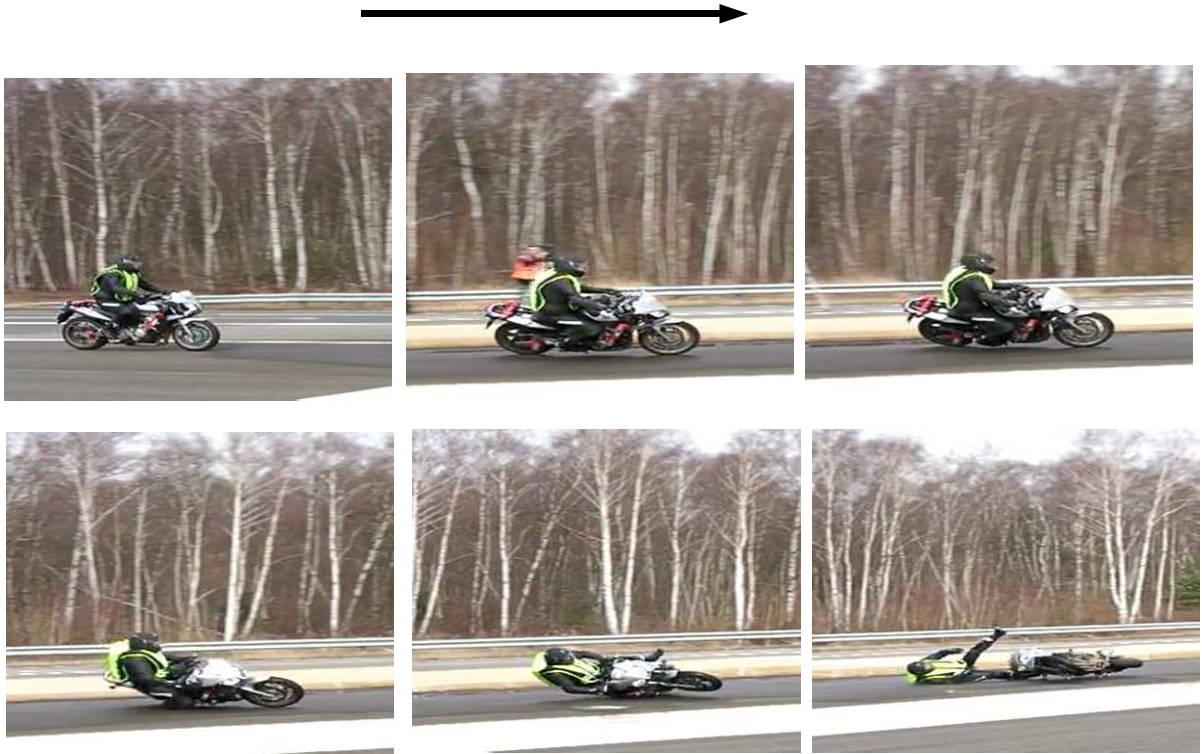


Figure 3.5: Les vidéos enregistrées avec la camera rapide (1000 images par seconde) au cours d'une séquence de chute.

instruction a été donnée au cascadeur de provoquer la chute au rond-point en glissant à une vitesse de 30 km / h.

Les cinq scénarios ont été reproduits par le cascadeur, une fois pour les scénarios 2 et 4, deux fois pour les scénarios 1, 3 et 5.

3.4 Construction de la base de données

Dans ces travaux, nous avons fait le choix d'utiliser uniquement des données issues des capteurs accéléromètre et gyromètre, afin d'atteindre nos objectifs. Ce choix est motivé à la fois, par le fait que ce type d'informations permet de détecter tous les changements dans la dynamique de la moto, et par la facilité de l'installation de ces capteurs sur le 2RM.

Pour simplifier la compréhension de la base de données collectée au cours des différentes expérimentations, celle-ci peut être décrite de la manière suivante:

$$\mathcal{D} = \langle \mathcal{D}^l \rangle_{l=1:\mathcal{L}} . \quad (3.1)$$

Avec $\mathcal{D}^l = \{\mathbf{x}_{j,k}^{(l)}\}$, \mathcal{L} est le nombre total de séquences. Pour chaque séquence $l \in \{1, \dots, \mathcal{L}\}$, il existe un vecteur $\mathbf{x}_k^{(l)}$ formé par les données acquises par un capteur j à l'instant t_k , avec $t_k \in [1, T^{(l)}]$, $T^{(l)}$ représente la durée totale de la séquence l , $j \in \{1, \dots, 6\}$, où $k \in [1, N^{(l)}]$, avec $N^{(l)}$ est le nombre d'individus de la séquence l . Ainsi, \mathbf{x}^l est défini comme suit:

$$\mathbf{x}^l = \{a_x, a_y, a_z, r_x, r_y, r_z\}, \quad (3.2)$$

avec,

- a_x, a_y et a_z représentent respectivement les accélérations longitudinales, latérales et verticales.
- r_x, r_y et r_z représentent respectivement les vitesses angulaires du roulis, du tangage et du lacet.

3.5 Pré-traitement de données

Les données de mesures issues des expérimentations réelles sont par nature incomplète et entachées d'incertitudes et de valeurs manquantes (bruit de hautes fréquences qui est causé principalement par les vibrations du moteur). Le pré-traitement des données s'avère alors une étape capitale en amont de l'étape de classification. Le remplacement des valeurs manquantes, le filtrage du bruit ainsi que la normalisation des données sont les différentes étapes constituant cette phase de pré-traitement.

3.5.1 Remplacement des valeurs manquantes et aberrantes

En premier lieu, nous nous sommes intéressés au remplacement des valeurs manquantes. Cette étape a nécessité un temps important d'analyse et d'investigation au cours duquel, nous avons découvert certaines particularités liées aux données traitées, des micro-coupures dans de l'acquisition des données qui sont probablement dues à des problèmes d'alimentation des capteurs. Cette étape a permis de définir un pré-traitement spécifique à notre problématique. Nous avons choisi de remplacer les valeurs manquantes par la moyenne empirique calculée sur un intervalle de 4s (2s avant et 2s après la coupure). Ce qui résulte d'une moyenne estimée sur $1000 \times 4 = 4000$ points.

3.5.2 Filtrage des données

En raison de la complexité et de la nature bruitée des données collectées, une étape de filtrage nous est apparue nécessaire. Pour ce faire, nous avons d'abord procédé à une analyse fréquentielle des signaux acquis afin de déterminer la fréquence utile des données

grâce à la TFDF (Transformée de Fourier Discrète Fenêtrée). Comme illustré par la figure 3.6, qui représente le spectrogramme de l'accélération longitudinale, nous avons constaté la présence d'un bruit quand la moto est à l'arrêt aux environs de 40 Hz. En réalité, ce bruit n'est pas lié au capteur, mais plutôt, un bruit qui est dû aux vibrations du moteur de la moto. Ces vibrations sont tout simplement celles du moteur et celles-ci augmentent ou diminuent en fonction du régime moteur. La présence de ce bruit dans les mesures s'explique par les faibles dimensions de la moto et la présence du moteur au centre de la moto, non loin de l'endroit où la centrale inertielle (accéléromètres et gyromètres) est placée. Nous avons également constaté la présence d'un bruit de haute fréquence de la forme de deux exponentielles croissante et décroissante, qui sont symétriques par rapport à l'axe parallèle à l'axe des temps à la fréquence 290 Hz. Ce bruit est probablement dû au bruit propre du capteur, ce dernier a été constaté sur tous les capteurs et sur tous les enregistrements.

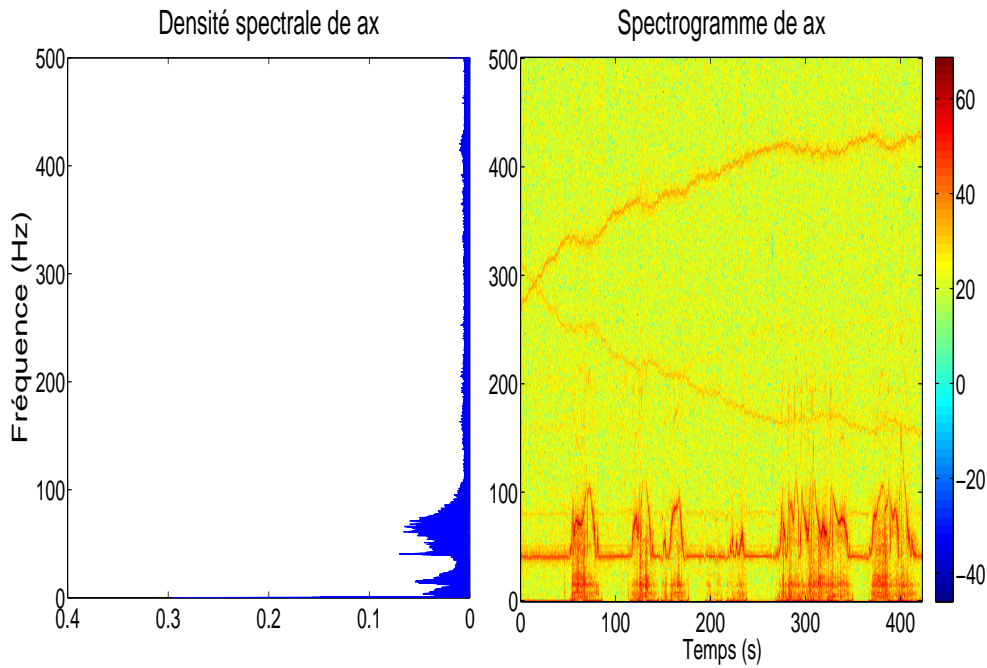


Figure 3.6: La densité spectrale de puissance de l'accélération longitudinale (ax) à gauche et son spectrogramme à droite de la figure.

Les variations des signaux sont très lentes, dues à la nature mécanique de la moto [4]. Pour cela, nous nous n'intéressons pas à des fréquences supérieures à 2 Hz, fréquences qui représentent le bruit qui pourrait causer des problèmes à nos expériences ultérieures. Nous proposons un filtrage de données en utilisant un filtre passe-bas pour éliminer ce bruit. Pour cela, nous avons implémenté trois types de filtres à savoir:

- Le filtre de Butterworth avec une fréquence de coupure de 2 Hz;
- Le filtre médian avec une fenêtre glissante de taille 10 échantillons;
- Le filtre à ondelettes avec un niveaux de décomposition égale à 6 et une ondelette mère de type (Db20).

Nous avons utilisé la valeur du RSB (Rapport Signal sur Bruit) et le coefficient de corrélation (R) de chaque filtre comme critère pour comparer leurs performances [8]. Le RSB est défini comme le rapport de la puissance du signal sur la puissance du bruit. Ainsi, pour un signal issu du capteur j provenant de la séquence de conduite l , le rapport signal sur bruit est exprimé ($RSB_j^{(l)}$) par l'équation suivante:

$$RSB_j^{(l)} = 10 * \log \frac{\sum_{k=0}^{N^{(l)}-1} |\mathbf{x}_{j,k}^{(l)}|^2}{\sum_{k=0}^{N^{(l)}-1} |\mathbf{x}_{j,k}^{(l)} - \mathbf{y}_{j,k}^{(l)}|^2}. \quad (3.3)$$

Le coefficient de corrélation ($R_j^{(l)}$) pour un signal j provenant de la séquence de conduite l est calculé de la manière suivante:

$$R_j^{(l)} = \frac{\sum_{k=0}^{N^{(l)}-1} \mathbf{x}_{j,k}^{(l)} \mathbf{y}_{j,k}^{(l)}}{\left[\sum_{k=0}^{N^{(l)}-1} (\mathbf{x}_{j,k}^{(l)})^2 \sum_{k=0}^{N^{(l)}-1} (\mathbf{y}_{j,k}^{(l)})^2 \right]^{\frac{1}{2}}}. \quad (3.4)$$

Avec $\mathbf{y}_k^{(l)}$ et $\mathbf{x}_k^{(l)}$ représentent respectivement le signal filtré et le signal de référence acquis par le capteur j au cours de la séquence de conduite l .

Le tableau 3.1 dresse les résultats obtenus des RSBs et Rs de chaque filtre et pour chaque signal. Comme nous pouvons le constater, le filtre médian donne de meilleurs résultats par rapport aux deux autres filtres. Ces résultats permettent de choisir le filtre médian pour filtrer les données des accéléromètres et des gyromètres.

La figure 3.7 illustre un exemple de données (accéléromètres et gyromètres) brutes (en bleu) et filtrées (en rouge) acquises au cours d'une séquence de conduite normale.

3.5.3 Annotation/Normalisation des données

Afin, de travailler dans le cadre d'un apprentissage supervisé, nous avons étiqueté nos données dans le but de construire les bases d'apprentissage et de test. L'étiquetage de ces données a été effectué manuellement grâce à un logiciel propriétaire BINASCII développé au sein de l'IFSTTAR. Ce logiciel permet de visualiser et de synchroniser les données des capteurs (vidéos et inertiels). La détermination des différentes plages temporelles de chaque situation de conduite (temps de début et fin) a été effectuée en visualisant les données vidéos au ralenti. Comme le montre la figure 3.8, les situations

Table 3.1: Comparaison des valeurs de RSB et de R des trois filtres utilisés sur les signaux d'accélération et de vitesses angulaires.

Critères d'évaluation	Le filtre à ondelettes	Le filtre médian	Le filtre de Butterworth
RSB (dB) de a_x	5.45 ± 0.34	5.54 ± 0.44	2.95 ± 2.8
R de a_x	0.59 ± 0.01	0.60 ± 0.01	0.46 ± 0.01
RSB (dB) de a_y	8.85 ± 1.06	12.06 ± 1.35	5.37 ± 0.90
R de a_y	0.59 ± 0.02	0.64 ± 0.01	0.43 ± 0.03
RSB (dB) de a_z	32.93 ± 0.68	34.41 ± 0.68	29.41 ± 0.54
R de a_z	0.97 ± 0.01	0.97 ± 0.02	0.96 ± 0.01
RSB (dB) de r_x	12.76 ± 1.04	13.75 ± 1.06	-0.8 ± 0.04
R de r_x	0.72 ± 0.02	0.74 ± 0.02	0.32 ± 0.02
RSB (dB) de r_y	9.05 ± 0.93	8.37 ± 0.93	4.42 ± 0.94
R de r_y	0.62 ± 0.02	0.59 ± 0.01	0.38 ± 0.03
RSB (dB) de r_z	17.12 ± 1.04	19.10 ± 1.05	6.20 ± 0.83
R de r_z	0.81 ± 0.19	0.84 ± 0.02	0.59 ± 0.02

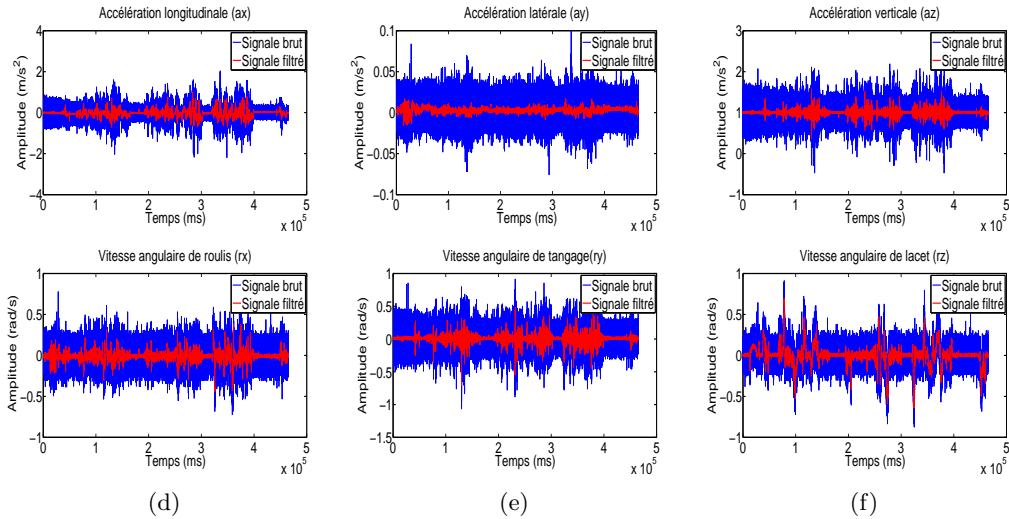


Figure 3.7: Représentation des signaux accélérométriques et gyroscopiques bruts (en bleu) et filtrés (en rouge) enregistrés au cours d'une séquence de conduite normale.

de conduite rencontrées au cours d'un de ces trajets: les arrêts (AR), les lignes droite (LD), les virages à gauche (VG), les virages à droite (VD), les ronds point (RP).

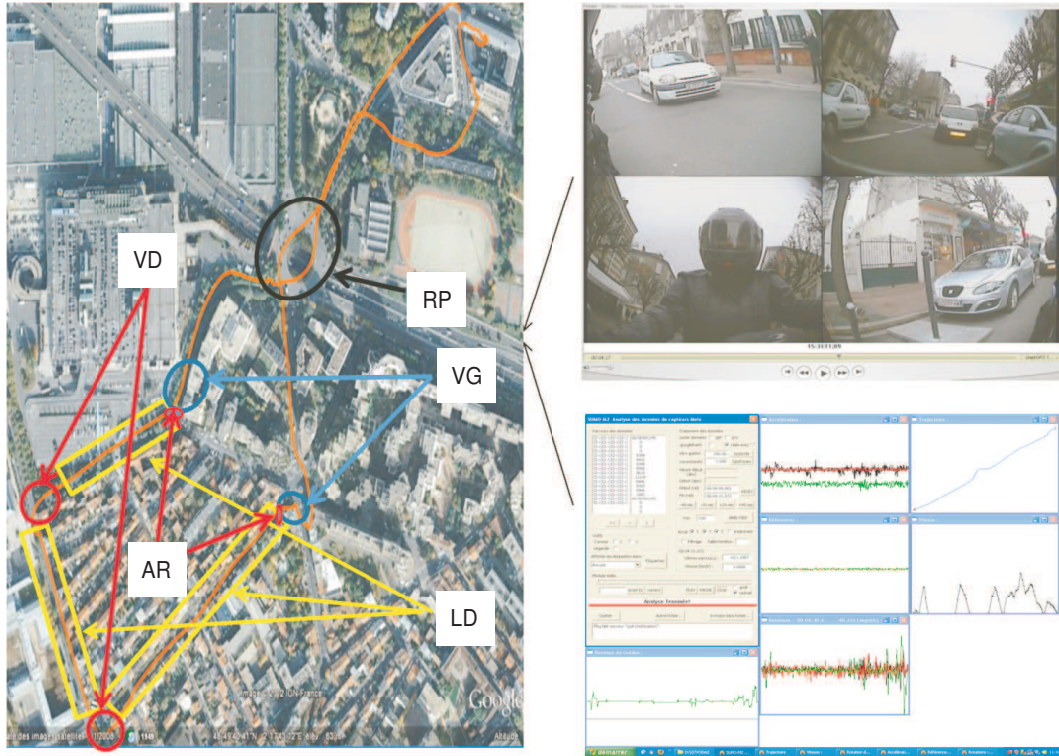


Figure 3.8: Annotation des données en utilisant le logiciel "BinAscii".

Ainsi, après l'annotation de la base d'apprentissage constituée des 11 séquences de conduite normale, on a identifié 151 situations LD, 81 situations VD, 42 situations VG et 20 situations RP. La figure 3.10 représente le nombre d'échantillons en pourcentage de chaque situation (classe) dans chaque séquence ainsi que le nombre total d'échantillons de chaque situation dans toute la base de données. Comme nous pouvons le constater, les VG sont faiblement représentés avec seulement 6,65% dans toute la base de données. Notons que la majorité des séquences est composée de LD avec 66,07%. Cela est dû à la trajectoire qui a été effectuée par des conducteurs qui contient de longues LD et de courts VG.

Après l'étape de filtrage, et vu que les amplitudes de variations des données étant différentes d'une variable à une autre, et pour éviter la domination d'un signal sur les autres dans l'étape de classification, nous proposons de normaliser les données de mesures. Pour cela, nous avons testé plusieurs types de normalisation (centrée réduite, centrée, max-min, ...). Nous avons décidé de centrer nos données étant donné que le résultat de classification est meilleur que les autres méthodes de normalisation.

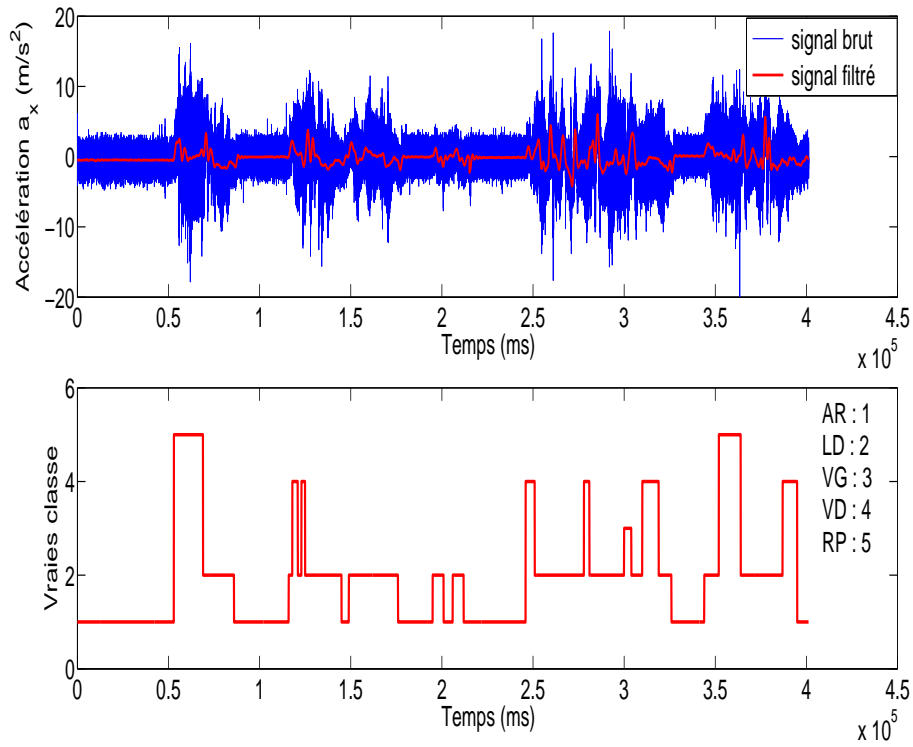


Figure 3.9: L'accélération longitudinale enregistrée au cours d'une séquence de conduite normale et les classes correspondantes.

3.6 Conclusion

Dans ce chapitre, nous avons décrit l'outil de recueil de données installé sur une moto. Nous avons décrit également le protocole expérimental ainsi que les séquences de conduite réalisées qui constituent notre base de données (séquence de conduite normale, séquence de conduite à risque et séquence de conduite avec chute) voir table 3.2. Pour la mise en forme de la base de données, une étape de pré-traitement a été effectuée. Cette étape a permis le traitement de valeurs manquantes, le filtrage des données à l'aide d'un filtre médian ainsi que l'annotation des séquences et la normalisation des données.

Table 3.2: Récapitulatif des différentes séquences de conduite réalisées

Séquences	Type de scénario	Scénarios	Temps de chute (ms)
Séquence 1	Conduite avec chute	Chute à l'arrêt	14080
Séquence 2	Conduite avec chute	Chute à l'arrêt	47070
Séquence 3	Conduite avec chute	Perte adhérence en virage	56380
Séquence 4	Conduite avec chute	Perte adhérence en virage	43750
Séquence 5	Conduite avec chute	Perte adhérence sur un rond point	36160
Séquence 6	Conduite avec chute	Couchage de la moto	34540
Séquence 7	Conduite avec chute	Couchage de la moto	34500
Séquence 8	Conduite avec chute	Perte adhérence en ligne droite	40420
Séquence 9	Conduite à risque	Conduite sportive	0
Séquence 10	Conduite à risque	Conduite sportive	0
Séquence 11	Conduite à risque	Conduite sportive	0
Séquence 12	Conduite à risque	Accélération brusque	0
Séquence 13	Conduite à risque	Accélération brusque	0
Séquence 14	Conduite à risque	Freinage brusque	0
Séquence 15	Conduite à risque	Conduite sur une piste	0
Séquence 16	Conduite à risque	Conduite sur une piste	0
Séquence 17	Conduite à risque	Perte de contrôle avec rattrapage	0
Séquence 18	Conduite à risque	Conduite en zigzag	0
Séquences 19-29	Conduite normale		0

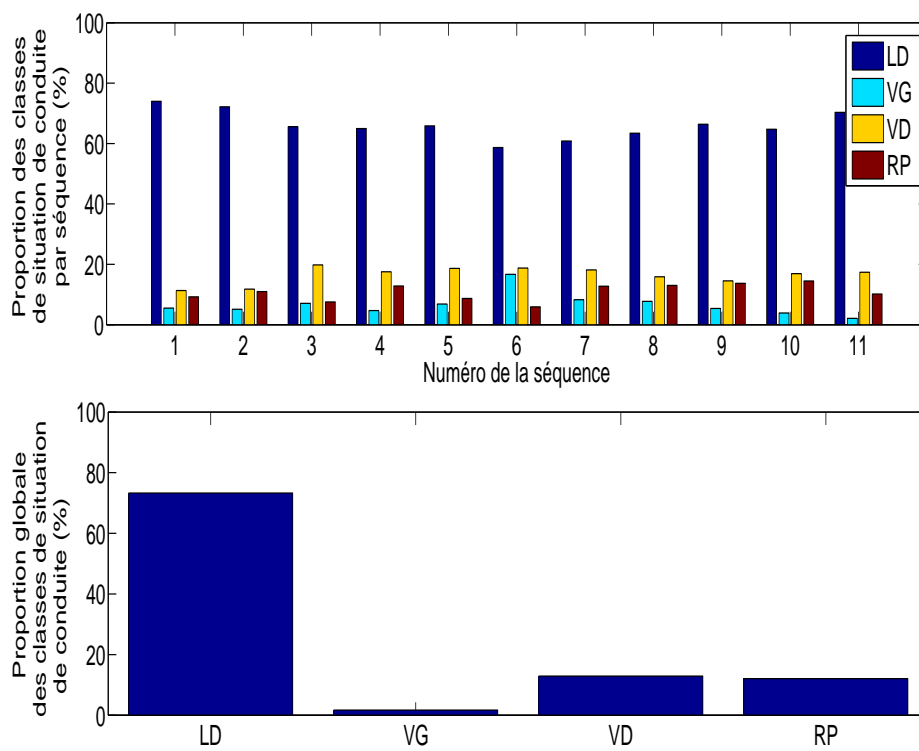


Figure 3.10: Représentation des proportions des classes de situation de conduite de chaque séquence de conduite normale (en haut) et pour toutes les séquences (en bas).

Chapitre 4

Éléments sur la théorie de l'apprentissage

Sommaire

4.1	Introduction	34
4.2	Généralités sur l'apprentissage	34
4.2.1	Apprentissage supervisé	34
4.2.2	Apprentissage non supervisé	37
4.3	Sélection des caractéristiques	38
4.3.1	Phase de sélection des meilleurs sous ensembles	39
4.3.2	Phase de validation	42
4.4	Évaluation des performances	42
4.4.1	Mesure de la performance de généralisation	42
4.4.2	Mesure de la performance d'un classifieur	44
4.5	Conclusion	48

4.1 Introduction

L'objectif de ce chapitre est de fournir au lecteur une vue d'ensemble des notions fondamentales intervenant lors de la mise en œuvre d'un système de reconnaissance de forme à base d'apprentissage statistique. Dans un premier temps, nous décrivons l'apprentissage supervisé tout en détaillant les deux approches discriminative et générative, ainsi que les deux types de problèmes traités par ce type d'apprentissage à savoir la classification et la régression. Une autre catégorie d'apprentissage sera décrite: l'apprentissage non supervisé. Pour améliorer les méthodes d'apprentissage, Nous présenterons quelques approches de base de sélection des caractéristiques: approche filtre, approche wrapper et approche hybride. Enfin, nous mettrons en exergue un ensemble de critères pour l'évaluation des performances des classifieurs.

4.2 Généralités sur l'apprentissage

Dans un cadre général, un apprentissage est l'acquisition de nouveaux savoirs (souvent sous forme de connaissances), c'est-à-dire le processus d'acquisition de compétences, par l'expérience ou l'enseignement [9]. L'apprentissage est un terme utilisé dans plusieurs domaines (philosophie, linguistique, psychologie, biologie, ...), mais chacun a sa propre définition de cette notion. Comme notre travail s'inscrit dans le domaine des sciences de l'ingénieur (informatique, automatique, robotique) et également dans le domaine de l'Intelligence Artificielle nous parlerons d'apprentissage automatique plutôt que d'apprentissage tout court.

Au cours des deux dernières décennies l'apprentissage automatique est devenu l'un des piliers de la technologie. Avec la croissance des quantités de données disponibles créées, stockées, manipulées, il y a de bonnes raisons de croire que l'analyse automatique des données deviendra encore plus omniprésente comme un ingrédient nécessaire pour le progrès technologique. Les domaines d'applications sont divers et multiples. Nous en citerons ici quelques-uns en Biologie [10, 11, 12], en Finance [13, 14] et dans l'industrie automobile [4, 15].

Dans ce qui suit, nous allons présenter les deux approches de l'apprentissage statistique, à savoir l'apprentissage supervisé et l'apprentissage non supervisé.

4.2.1 Apprentissage supervisé

Dans la littérature, l'apprentissage supervisé est défini de plusieurs façons. Parmi ces définitions, on trouve celle donnée par [9] qui postule que l'apprentissage supervisé peut être caractérisé par trois éléments: i) la base d'apprentissage, ii) la tâche, iii) la mesure de performance.

L'apprentissage supervisé fait intervenir les éléments suivants [16]:

- Des variables d'entrées ($\mathbf{X} \in \mathcal{X}$) appelées aussi covariables appartenant à l'ensemble \mathbb{R} des réels de dimension p , noté \mathbb{R}^p .
- Des variables de sorties ($\mathbf{Y} \in \mathcal{Y}$) appelées aussi variables d'intérêts appartenant à l'ensemble des réels \mathbb{R} . Les variables d'intérêts peuvent être continues (problème de régression) ou discrètes, variables catégorielles (problème de classification).
- Une fonction de prédiction notée $f_{\boldsymbol{\theta}}(\mathbf{x})$ à paramètre $\boldsymbol{\theta}$ appartenant à l'ensemble des paramétrés Θ . Cette fonction a pour rôle d'associer des variables de l'espace d'entrée \mathbf{X} avec des variables de l'espace de sortie \mathbf{Y} .
- Un ensemble d'apprentissage $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ constitué par l'ensemble de réalisation du couple $(\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3), \dots, (\mathbf{x}_N, y_N)\}$ supposées indépendantes et identiquement distribuées (i.i.d) issues de la loi de probabilité jointe inconnue $p(\mathbf{x}; y) = p(\mathbf{x})p(y|\mathbf{x})$.
- Un ensemble de test constitué par l'ensemble de nouvelles réalisations de X , et l'objectif est de prédire la variable $y = f_{\boldsymbol{\theta}}(\mathbf{x})$.
- Une fonction coût de l'erreur notée $L(f_{\boldsymbol{\theta}}(\mathbf{x}), y)$ permettant de traduire l'erreur d'attribution de la variable d'entrée à sa variable de sortie. La forme de cette fonction dépend essentiellement du problème étudié.

L'apprentissage supervisé est un cadre de recherche de la relation entre les attributs d'entrées et de sorties à partir d'une base $\mathcal{S} \subset (\mathcal{X} \times \mathcal{Y})$ dite base d'apprentissage définie par:

$$\mathcal{S} \triangleq \{(\mathbf{x}_i, y_i)\}_{i=1}^N. \quad (4.1)$$

Dans l'apprentissage supervisé, l'objectif est d'estimer la fonction de prédiction $f_{\boldsymbol{\theta}}(\mathbf{x})$ qui minimise le coût de l'erreur (ou fonction de perte¹) pour les futures réalisations du couple (\mathbf{X}, \mathbf{Y}) ainsi que le risque total moyen défini par :

$$\mathbb{E}[L(f_{\boldsymbol{\theta}}(\mathbf{x}), y)] = \int_{\mathcal{X}} \int_{\mathcal{Y}} L(f_{\boldsymbol{\theta}}(\mathbf{x}), y) p(\mathbf{x}, y) d\mathbf{x} dy \quad (4.2)$$

Le calcul du risque moyen donné par l'équation (4.2) suppose que la loi jointe du couple (\mathbf{X}, \mathbf{Y}) est connue. Malheureusement, en pratique cette loi n'est pas connue et la seule connaissance qu'on a sur le problème est relative aux valeurs de ces réalisations. Cependant, le risque empirique est calculé en remplaçant l'espérance par une moyenne

¹En anglais "loss" function.

empirique. Formellement, le risque empirique est défini comme suit:

$$R_{emp}(f_{\theta}(\mathbf{x})) = \frac{1}{N} \sum_{i=1}^N [L(f_{\theta}(\mathbf{x}_i), y_i)]. \quad (4.3)$$

Comme indiqué précédemment, l'apprentissage supervisé traite souvent deux types de problèmes liés essentiellement à la nature de la sortie \mathbf{Y} , à savoir le problème de classification et le problème de régression.

Le problème de la classification

On parle de problème de classification quand la variable d'intérêt y appartient à un ensemble fini $Y = \{c_1, c_2, c_3, \dots, c_k\}$. En d'autres termes le but de la classification est de prédire l'appartenance d'un objet x décrit par le vecteur de caractéristiques à une classe donnée $\{c_i\}_{i=1}^k$. Dans ce cas de figure, la fonction de coût L prend souvent la forme suivante:

$$L(f_{\theta}(\mathbf{x}), y) \triangleq \begin{cases} 0 & \text{si } f_{\theta}(\mathbf{x}) = y, \\ 1 & \text{sinon.} \end{cases} \quad (4.4)$$

Le problème de la régression

Dans le problème de régression y est une variable continue qui appartient à l'ensemble des réels \mathbb{R} . Le but de la régression est d'associer une valeur numérique à une réalisation de la variable d'entrée \mathbf{x} . La fonction de coût L , la plus utilisée dans la littérature est de la forme suivante:

$$L(f_{\theta}(\mathbf{x}), y) = (f_{\theta}(\mathbf{x}) - y)^2. \quad (4.5)$$

Approches génératives/discriminatives

Dans l'apprentissage statistique, on distingue deux groupes d'approches à savoir les approches génératives et les approches discriminatives. Dans l'approche générative, l'algorithme apprend un modèle à partir de la distribution conjointe $p(\mathbf{x}, y)$. Ce modèle consiste à modéliser pour tout $\mathbf{Y} = \{c_1, c_2, c_3, \dots, c_k\}$, les densités conditionnelles $p(\mathbf{x}|y)$ ainsi que la probabilité *a priori* $p(y)$. Les probabilités *a posteriori* de chaque classe sont ensuite calculées en utilisant le théorème de Bayes:

$$p(\mathbf{x}|y) = \frac{p(\mathbf{x})p(y|\mathbf{x})}{\sum_{y'} p(y')p(y|y')}. \quad (4.6)$$

L'approche discriminative modélise directement la distribution conditionnelle $p(y|\mathbf{x})$. A partir de cette distribution conditionnelle, nous pouvons faire des prédictions de y , pour toute nouvelle valeur de \mathbf{x} , en utilisant la règle du Maximum A Posteriori (MAP):

$$\hat{y} = \operatorname{argmax}_{y \in Y} p(y|\mathbf{x}). \quad (4.7)$$

4.2.2 Apprentissage non supervisé

L'apprentissage non supervisé consiste à chercher un partitionnement homogène d'une manière qui reflète la structure statistique de l'ensemble des motifs d'entrées [17]. Contrairement à l'apprentissage supervisé, dans l'apprentissage non supervisé, il n'y a pas de variables de sortie associée à chaque variable d'entrée \mathbf{X} .

L'algorithme de l'apprentissage non supervisé apprend à partir d'une base d'apprentissage définie de la façon suivante:

$$\mathcal{S} = \{(\mathbf{x}_i)\}_{i=1}^N. \quad (4.8)$$

Les problèmes traités par l'apprentissage non supervisé sont: le regroupement automatique et la réduction de la dimension. Les solutions proposées pour résoudre ces deux problèmes s'appuient généralement sur des arguments géométriques ou sur une modélisation probabiliste.

Regroupement automatique

Le problème du regroupement automatique (ou le Clustering en anglais) consiste à regrouper les données en sous-groupes ayant des caractéristiques statistiques semblables. Autrement dit, le but est de séparer les données d'entrées en groupes d'individus qui présentent des caractéristiques similaires. Le regroupement automatique est riche par ces méthodes (les approches hiérarchique [18] (Classification Ascendante Hiérarchique, CAH), l'algorithme des k-means [19], les modèles de mélange [20], ...). L'intérêt de cette méthode réside dans le fait qu'il permet d'avoir les mêmes sorties qu'un apprentissage supervisé en se basant juste sur les variables d'entrées.

Réduction de la dimension

Dans les problématiques de reconnaissance de formes, lorsque la dimension de l'espace des variables d'entrées d'un système est élevée, il est important et nécessaire de réduire cette dimension. En effet, les variables décrivant le problème à modéliser ne sont pas toutes de la même importance pour la compréhension du phénomène étudié. Il existe des variables qui sont pertinentes pour l'identification du modèle et d'autres qui ne le sont pas.

Toutefois, la réduction de la dimension de l'espace des variables d'entrées a pour objectifs de faciliter l'analyse des données, d'améliorer les performances des classificateurs, de sélectionner les ensembles de données les plus cohérents à la fois pour plus de précision et pour la réduction du temps de calcul des algorithmes, enlever les informations redondantes ou non pertinentes, etc ... Les méthodes de la réduction de la dimensionnalité sont généralement regroupées en deux approches: les approches linéaires (ACP Analyse en Composantes Principales, ALD Analyse Linéaire Discriminante, MDS Multi-Dimensional Scaling) et les approches non linéaires (Isomap, LLE (Locally Linear Embedding)), pour plus de détail voir [21].

4.3 Sélection des caractéristiques

La sélection des caractéristiques est un terme couramment utilisé dans le domaine de la fouille de données. Elle désigne souvent le processus de recherche d'un sous-ensemble "pertinent" de caractéristiques parmi celles de l'ensemble initial. La sélection des caractéristiques implique non seulement la réduction de la cardinalité de l'ensemble des variables d'intérêt, mais aussi le choix des attributs les plus pertinents.

Une description formelle de la sélection des caractéristiques peut être exposée comme suit: soit $\mathcal{F} = \{f_1, f_2, \dots, f_p\}$ un ensemble de caractéristiques de taille p , où p est le nombre total de caractéristiques. On définit une fonction E_v qui évalue la pertinence des sous-ensembles sélectionnés et cette fonction est maximale pour le "meilleur" sous-ensemble. Par conséquent, l'objectif de la sélection de p caractéristiques est de trouver un sous-ensemble optimal \mathcal{F}' avec $(\mathcal{F}' \subset \mathcal{F})$ de taille p' avec $(p' \leq p)$ de telle sorte que l'égalité suivante soit vérifiée:

$$E_v(\mathcal{F}') = \arg\max_{Z \subset \mathcal{F}} E_v(Z). \quad (4.9)$$

Avec $\|Z\| = p'$ et p' est un nombre prédéfini par l'utilisateur ou bien contrôlé par le critère utilisé dans le processus de sélection.

Un processus de sélection de caractéristiques peut être structuré en deux phases: la phase de sélection des meilleurs sous-ensembles, et la phase de validation comme illustré sur la figure 4.1 [22].

- La phase de sélection est subdivisée en trois étapes: la génération d'un sous-ensemble à partir de l'ensemble initial selon une stratégie de recherche, qui peut être complète, séquentielle ou aléatoire,
- La phase d'évaluation de tous les sous-ensemble générés dans la première phase selon un critère d'évaluation, qui peut être filtre, enveloppe ou hybride.

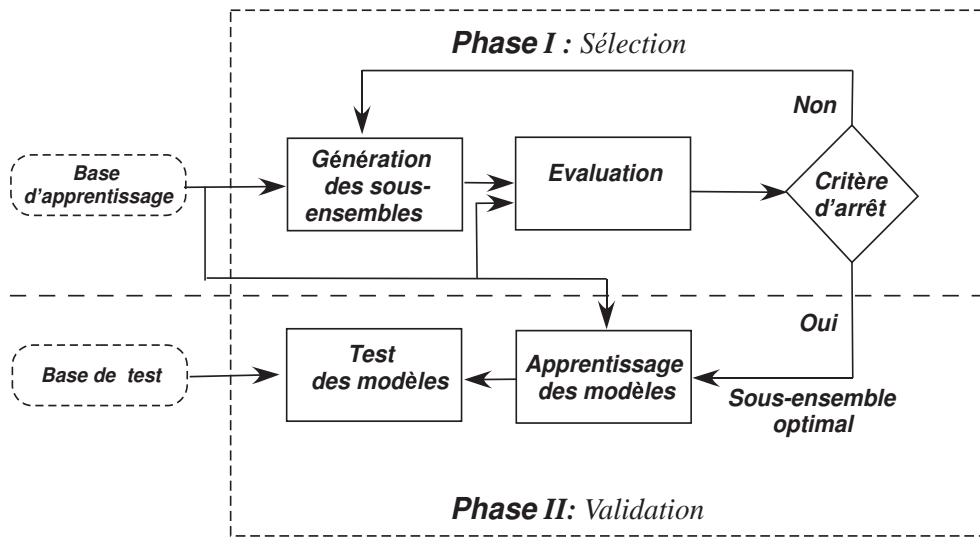


Figure 4.1: Processus de sélection des caractéristiques.

4.3.1 Phase de sélection des meilleurs sous ensembles

Stratégies de recherche

- **Stratégie exhaustive**: elle consiste à la génération de toutes les combinaisons possibles qui peuvent être formées avec les éléments de l'ensemble initial de caractéristique \mathcal{F} . Cette stratégie de recherche permet de trouver le sous-ensemble optimal \mathcal{F}' . cependant, l'inconvénient de cette méthode est le nombre de sous-ensembles générés qui croît exponentiellement en fonction du nombre total de caractéristiques à sélectionner. Ainsi, pour un ensemble de caractéristiques de dimension p , le nombre total de sous-ensembles générés, induirait une complexité équivalente à $O(2^p)$, qui peut être très coûteux en temps de calcul.
- **Stratégie heuristique**: elle est basée sur des techniques de recherche qui fonctionnent selon des règles empiriques et visent à réduire la complexité de calcul sans dégrader les performances des classifieurs. En outre, cette stratégie exige une condition d'arrêt pour empêcher que la recherche des sous-ensembles ne devienne exhaustive. La génération de sous-ensembles se fait à l'aide d'une recherche itérative guidée par une heuristique qui peut être ascendante, descendante ou bien descendante-ascendante:
 1. **Ascendante** (ou *forward* en anglais): cette procédure de recherche se fait itérativement à partir d'un ensemble vide, à chaque itération, des variables seront rajoutées jusqu'à satisfaction d'un critère d'arrêt.
 2. **Descendante** (ou *backward* en anglais): cette procédure de recherche est presque similaire à l'approche ascendante à la seule différence, la recherche

commence à partir de l'ensemble de toutes les caractéristiques. A chaque itération, des variables seront rejetées.

3. Descendante-ascendante (ou *stepwise* en anglais) : Cette approche est la combinaison des deux approches précédentes. A chaque itération, des variables seront ajoutées ou rejetées.
- **Stratégie aléatoire**: elle permet de générer un nombre fini de sous-ensembles aléatoirement à partir de l'ensemble initial. La génération de ces sous-ensembles peut se faire de deux manières différentes. La première consiste à suivre la recherche heuristique par injection de l'aléatoire dans les approches heuristiques classiques décrites ci-dessus comme la méthode du recuit simulé. La deuxième consiste à générer les sous-ensembles d'une manière complètement aléatoire indépendamment les uns des autres (c'est-à-dire, la génération d'un sous-ensemble ne se fait pas par ajout ou suppression de variable à partir d'un sous-ensemble donné). L'utilisation du caractère aléatoire dans la génération des sous ensembles permet d'éviter les optima locaux dans l'espace de recherche, ce qui n'est pas le cas des méthodes heuristiques.

L'évaluation

Les trois procédures décrites précédemment permettent de générer des sous-ensembles de caractéristiques à sélectionner. Puis à l'aide de choix probabilistes basés sur des algorithmes ou des mesures de cohérence des données, des sous-ensembles optimaux peuvent être sélectionnés selon une fonction d'évaluation E_v . Cette fonction permet d'évaluer les sous-ensembles de caractéristiques à l'aide de certaines métriques. Les fonctions d'évaluation des sous-ensembles de caractéristiques générées sont regroupées en trois approches:

- **Approche filtre** (ou *Filter* en anglais) l'approche filtre repose sur l'analyse et l'évaluation des caractéristiques générales des données sans la mise en jeu d'un algorithme d'apprentissage. Cette approche est considérée comme un pré-traitement de données et intervient en aval de l'algorithme d'apprentissage comme illustré sur la figure 4.2. Les algorithmes de l'approche filtre se fondent sur un

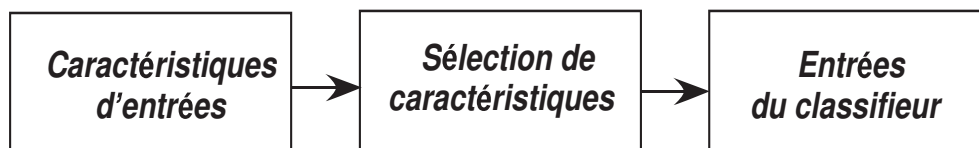


Figure 4.2: Approche Filtre.

critère de sélection (critère de Fisher, critère de corrélation, information mutuelle

...) qui reposent sur les propriétés des caractéristiques générées selon une stratégie de recherche (souvent heuristique).

En faisant varier la stratégie de recherche et le critère d'évaluation utilisés, différents algorithmes de sélection de caractéristiques de type filtre peuvent être créés. On peut citer la sélection séquentielle constructive SFS (Sequential Forward Selection) qui est l'un des premiers algorithmes de sélection de caractéristiques proposé par [23]. A partir d'un ensemble vide et une stratégie de recherche heuristique, l'algorithme sélectionne le meilleur sous-ensemble. Un autre algorithme similaire au SFS est la sélection séquentielle destructive SBS (Sequential Backward Selection) proposé par [24]. A la différence de l'algorithme SFS, dans l'algorithme SBS la recherche commence à partir de l'ensemble complet des caractéristiques. Puis la généralisation de ces deux algorithmes a été proposé, voir [25, 26]. On peut citer aussi l'algorithme focus [27] basé sur une recherche exhaustive à partir de l'ensemble de caractéristiques complet. L'un des algorithmes les plus connus et le plus utilisé est l'algorithme Relief [28, 29]. Cette algorithme est basé sur la mesure de la pertinence d'une caractéristique par rapport à la variable d'intérêt. L'algorithme relief réordonne les caractéristiques de l'ensemble initial selon leur appartenance en leur associant un degré de pertinence.

- **Approche enveloppe** (ou *Wrapper* en anglais): l'approche enveloppe est très similaire à l'approche filtre sauf que dans l'approche enveloppe, un algorithme d'apprentissage est utilisé à la place d'un critère de sélection pour l'évaluation du sous-ensemble généré (voir figure 4.3). Chaque sous-ensemble généré à partir de l'ensemble initial est évalué par un algorithme d'apprentissage. Par conséquent, on peut avoir différents résultats de sélection selon l'algorithme d'apprentissage utilisé. Plusieurs algorithmes de sélection peuvent être créés en faisant varier la stratégie de recherche et les algorithmes d'apprentissage mis en jeu pour l'évaluation des sous-ensembles générés.

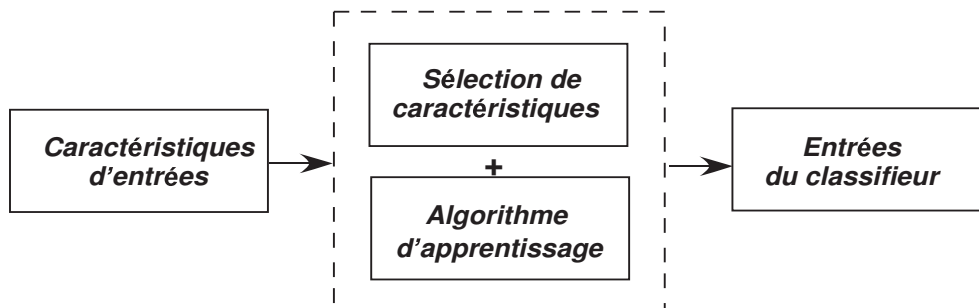


Figure 4.3: Approche wrapper et hybride.

- **Approche Hybride** (ou *Embedded* en anglais): Pour profiter des avantages des deux approches décrites ci-dessus, l'approche hybride a été récemment pro-

posée pour traiter des ensembles de données volumineux [30]. Les algorithmes de l'approche hybride utilisent à la fois un critère de sélection et un algorithme d'apprentissage pour évaluer les sous-ensembles générés. Le critère de sélection est utilisé pour sélectionner les meilleurs sous-ensembles générés selon une stratégie de recherche ensuite, l'algorithme d'apprentissage est utilisé pour sélectionner le meilleur sous-ensemble parmi les meilleurs sous-ensembles sélectionnés.

Critère d'arrêt

Un critère d'arrêt détermine le moment où le processus de sélection des caractéristiques doit s'arrêter. Selon l'application et l'objectif visé, le critère d'arrêt peut être de diverses natures : un nombre minimum de caractéristiques ou nombre maximum d'itérations, un taux de bonne classification, un temps de calcul, etc

4.3.2 Phase de validation

Dans les applications réelles, on ne dispose pas souvent de connaissances a priori sur les données (les variables pertinentes). Par conséquent, des méthodes indirectes peuvent être utilisées pour évaluer la pertinence des caractéristiques sélectionnées. Par exemple, le taux d'erreur de classification peut être utilisé comme indicateur de performance pour la validation des sous-ensembles sélectionnés. Les différentes méthodes d'évaluation des performances des algorithmes d'apprentissage seront décrites dans la section suivante.

4.4 Évaluation des performances

4.4.1 Mesure de la performance de généralisation

La méthode de validation est une méthode de vérification *a posteriori* des performances d'un modèle. La question de la sélection de modèle est abordée dans le sens de la sélection de l'architecture optimale, c'est-à-dire de la complexité de la famille de fonctions utilisées. On construit un nombre de modèles de structures différentes. Ensuite, on réalise l'apprentissage des paramètres optimaux θ pour chacun d'entre eux. Puis, on sélectionne la structure (à travers ces paramètres optimaux) qui donne la plus faible erreur de généralisation. Il est crucial de mesurer l'erreur de généralisation sur des exemples qui n'ont pas servi lors de la construction du modèle. Pour cela, on divise l'ensemble des données disponibles en deux parties :

- un sous-ensemble d'apprentissage (\mathcal{S}) dont les données serviront à la construction du modèle ;
- un sous-ensemble de test (\mathcal{V}) dont les données seront utilisées uniquement pour évaluer la performance du modèle construit.

A partir d'un ensemble d'échantillons de taille N , il existe plusieurs méthodes (appelées techniques de rééchantillonnage) pour estimer la qualité de l'apprentissage. Ces méthodes sont le plus souvent appliquées avec l'hypothèse que la base de donnée utilisée est constituée de réalisations i.i.d de $p(\mathbf{x}, y)$. Parmi ces méthodes, on peut citer:

1. **Validation simple ou Hold out** : Cette technique de validation consiste à diviser les données disponibles en deux sous-ensembles (la base d'apprentissage (\mathcal{S}) et la base de test (\mathcal{V})) sans qu'aucune donnée ne soit commune. Un nombre assez conséquent de données est nécessaire dans l'ensemble de test pour estimer l'erreur de généralisation avec une bonne précision, réduisant d'autant le nombre de données disponibles pour l'apprentissage (voir [31]). Souvent on garde (70%) des données pour le sous-ensemble d'apprentissage et (30%) pour le test. Pour éviter autant que possible tout problème lié à une dérive des données (et donc à la remise en cause de leur caractère i.i.d.), les ensembles d'apprentissage et de test sont tirés aléatoirement de façon exhaustive parmi les N données disponibles.
2. **K-validation croisée (*K-fold cross validation*)** : l'échantillon initial est partitionné en K sous-ensembles disjoints de tailles approximativement identiques (N/K) [32]. On utilise tour à tour chacun de ces sous-ensembles comme jeu de test alors que les $(K - 1)$ autres servent pour l'apprentissage. Le modèle optimal est alors défini comme étant celui qui présente le meilleur score (erreur de généralisation minimum) de validation croisée.
3. **K validation croisée stratifiée (*stratified K- fold cross validation*)** : plusieurs travaux ont introduit le schéma de ré-échantillonnage stratifié en validation croisée. L'objectif est de respecter la répartition des classes dans chaque subdivision. L'idée sous-jacente est de réduire la variabilité des modèles produits lors de chaque passage. Néanmoins, certains auteurs [33] pensent, que cette stratégie n'est véritablement efficace que si l'échantillon initial a été extrait de manière stratifiée dans la population, c'est-à-dire que l'on a respecté de manière explicite les probabilités d'occurrence de chaque classe lors de la construction de l'échantillon.
4. **Leave One Out** : cette méthode est un cas particulier de la validation croisée pour lequel $K = N$ (voir [33, 34]). Cette technique nécessite de relancer N fois la méthode de classification sur $(N - 1)$ échantillons la rendant prohibitive en temps de calcul mais nécessaire quand le nombre d'échantillons n'est pas élevé.
5. **Bootstrap** : le Bootstrap [35] est une méthode relativement récente comparée aux autres méthodes citées ci-dessus. Cette méthode a été initialement développée afin d'estimer certains paramètres statistiques comme la moyenne, la variance, *etc.* Dans le cas de la sélection de modèles, le paramètre qui est estimé est

l'erreur de généralisation. Cependant, ce qui est particulier au Bootstrap, c'est que l'on n'estime pas l'erreur de généralisation directement, mais la différence entre l'erreur de généralisation et l'erreur d'apprentissage calculée sur un sous-ensemble de données de l'ensemble initial (tirage avec remise). Cette différence est appelée *optimisme*.

4.4.2 Mesure de la performance d'un classifieur

Matrice de confusion: Dans le domaine de l'apprentissage statistique, une matrice de confusion est un tableau de contingence qui permet la visualisation des performances d'un algorithme d'apprentissage. Elle permet aussi de voir facilement, si l'algorithme d'apprentissage confond deux classes. Chaque colonne de la matrice de confusion représente les instances d'une classe estimée tandis que chaque ligne représente des instances d'une classe réelle, voir 4.1.

Table 4.1: Matrice de Confusion dans le cas d'une classification binaire.

		classes obtenues	
		Positive	Négative
Vraies classes	Positive	V_p	F_n
	Négative	F_p	V_n

A partir de la matrice de confusion, plusieurs métriques d'évaluations peuvent être calculées :

Le taux de bonne classification (TBC): c'est la métrique la plus courante pour l'évaluation des performances des algorithmes d'apprentissage automatique. Elle mesure la proportion d'exemples classés correctement. Dans un problème de classification binaire, le taux de bonne classification est définie par:

$$TBC = \frac{V_p + V_n}{V_p + V_n + F_p + F_n}, \quad (4.10)$$

avec

- V_p représente les vrais positifs;
- V_n représente les vrais négatifs;
- F_n représente les faux négatifs;
- F_p représente les faux positifs.

Dans le cas où les classes ne sont pas équiréparties, le taux de bonne classification peut fournir une indication biaisée sur les performances du classifieur. Pour pallier

ce problème, d'autres critères peuvent être utilisés comme ceux que l'on évoquera ci-dessous:

Le rappel: c'est est une métrique qui permet de mesurer la capacité d'un classifieur à donner toutes les solutions pertinentes. L'expression de cette métrique est la suivante:

$$rappel = \frac{V_p}{V_p + F_n}. \quad (4.11)$$

La précision: la précision est une métrique qui permet de mesurer la capacité du classifieur à ne donner que les solutions pertinentes. L'expression de cette métrique est la suivante:

$$Precision = \frac{V_p}{V_p + F_p}. \quad (4.12)$$

La F-mesure : dans l'analyse statistique d'un classifieur binaire, la F-mesure est une métrique de performance basée sur la moyenne de la précision et du rappel. En d'autres termes elle mesure la capacité du classifieur à donner toutes les solutions pertinentes et à refuser les autres. La formule explicite générale de la F-mesure est donnée comme suit:

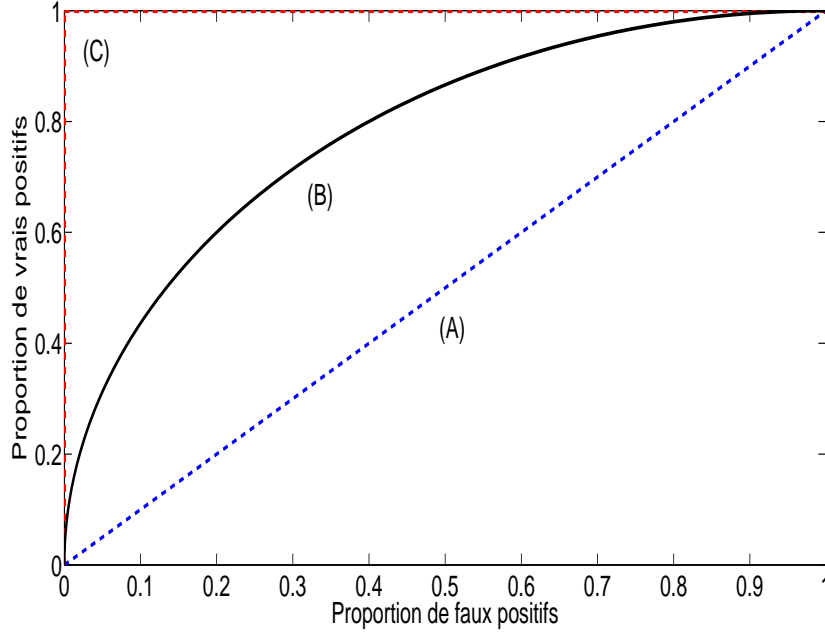
$$F_{\beta}mesure = \frac{(1 + \beta^2).rappel.precision}{\beta^2rappel + precision}. \quad (4.13)$$

avec β un facteur qui contrôle le degré d'importance de rappel/précision. β prend ses valeurs dans \mathbb{R}^+ .

La courbe ROC (Receiver Operating Characteristics): C'est une méthode graphique qui permet de visualiser les performances des algorithmes d'apprentissage statistique. Sur cette courbe est représentée en abscisse la proportion de faux négatifs et en ordonnée la proportion de vrais positifs voir figure 4.4.

- La courbe ROC idéale est représentée par la courbe (C). Dans ce cas de figure, l'algorithme prédit parfaitement toutes les étiquettes sans jamais se tromper. Un bon système d'aide à la décision essayera de tendre vers cette limite.
- Une courbe (B) représente un classifieur qui arrive à prédire les bonnes étiquettes en se trompant dans certains cas.
- Une courbe diagonale (la courbe(A)) représente le classifieur aléatoire, c'est un cas non informatif car les chances sont équiprobables pour que le classifieur arrive à prédire les bonnes étiquettes ou à se tromper.
- Si on obtient une courbe sous la diagonale, cela est souvent synonyme d'un problème sur les sorties du classifieur (présence d'un biais).

Un classifieur sera donc d'autant meilleur que sa courbe se situera proche de la courbe (C) et loin de la diagonale. Souvent, à partir de cette courbe, le critère (*AUC*) (aire sous la courbe ROC) est calculé. Ce critère permet de connaître le lien entre le taux de


 Figure 4.4: Exemples de modèle de courbes *ROC*.

vrais positifs et le taux de faux positifs. Plus, l'aire est importante, plus le classifieur possède un pouvoir discriminant sur les classes. L'utilisation du critère (*AUC*) peut se révéler fort intéressante dans le cas où une classe est sous représentée par rapport à une autre.

Enfin, pour mesurer le degré d'accord entre les prédictions des différents classifieurs et les vraies classes, nous avons utilisé la mesure kappa (κ_c) de Cohen [36]. Pour définir cet indice, nous commençons par définir la matrice de confusion suivante:

$$\left(\begin{array}{c} \text{Classes obtenues} \\ \text{Vraies classes} \end{array} \begin{array}{ccccccc} n_{11} & n_{12} & n_{13} & \dots & n_{1l} & \dots & n_{1m} \\ n_{21} & n_{22} & n_{23} & \dots & n_{2l} & \dots & n_{2m} \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\ n_{l1} & n_{l2} & n_{l3} & \dots & n_{ll} & \dots & n_{lm} \end{array} \begin{array}{c} \text{Total} \\ n_{.1} \\ n_{.2} \\ n_{.3} \\ \dots \\ n_{.l} \\ \dots \\ n_{.m} \\ n \end{array} \right)$$

l'indice kappa (κ_c) est donné par:

$$\kappa_c = \frac{p_a - p_e}{1 - p_e}, \quad (4.14)$$

avec

$$p_a = \frac{1}{n} \sum_{l=1}^m n_{ll}, \quad (4.15)$$

et,

$$p_e = \frac{1}{n^2} \sum_{l=1}^m n_{.l} n_{l.}. \quad (4.16)$$

Pour interpréter les résultats des valeurs de κ_c obtenues, on se base sur le tableau 4.2 proposé par [37].

Table 4.2: Interprétation des valeurs de κ_c

κ_c	Interprétation
< 0	Désaccord
$0.0 - 0.20$	Accord très faible
$0.21 - 0.40$	Accord faible
$0.41 - 0.60$	Accord modéré
$0.61 - 0.80$	Accord fort
$0.81 - 1.00$	Accord presque parfait

Afin de comparer statiquement les résultats des différents classifieurs, nous avons utilisé le test de McNemar's [38]. Ce test de McNemar non paramétrique est appliqué sur une table de contingence de dimension (2 x 2) (voir tableau 4.3) pour analyser des paires de données.

Table 4.3: Table de contingence

	$j^{\text{ème}}$ algorithme a bien réussi	$j^{\text{ème}}$ algorithme s'est trompé
$i^{\text{ème}}$ algorithme a réussi		N_{ij}
$i^{\text{ème}}$ algorithme s'est trompé	N_{ji}	

Ce test est basé sur la statistique du test de χ^2 qui est définie par:

$$\chi^2 = \frac{(\|N_{ij} + N_{ji}\| - 1)^2}{N_{ij} + N_{ji}}, \quad (4.17)$$

ainsi le Z score peut être facilement calculé à partir de l'équation (4.17).

$$Z_{ij} = \frac{|N_{ij} + N_{ji}| - 1}{\sqrt{N_{ij} + N_{ji}}}, \quad (4.18)$$

avec N_{ij} représente le nombre d'individus bien classés par le $i^{\text{ème}}$ algorithme mais mal classés par le $j^{\text{ème}}$ algorithme . N_{ji} représente le nombre d'individus bien classés par le $j^{\text{ème}}$ algorithme mais mal classés par le $i^{\text{ème}}$. Le $i^{\text{ème}}$ classifieur est statiquement mieux que le $j^{\text{ème}}$ classifieur si $|Z_{ij}| > 1.6$. Toutefois, la valeur 1.6 représente la valeur de la statistique pour une p -value à 5%.

4.5 Conclusion

Dans ce chapitre, nous avons introduit les notions de base de l'apprentissage statistique (apprentissage supervisé et apprentissage non supervisé). Nous avons aussi abordé le problème de sélection des caractéristiques. Les différentes méthodes de génération de sous-ensembles à sélectionner (exhaustive, séquentielles et aléatoire) ainsi que les approches d'évaluation (filtre, wrapper et hybride) de ces dernières ont été présentées. Pour finir nous avons présenté plusieurs méthodes destinées à évaluer l'erreur de généralisation d'un algorithme d'apprentissage.

Chapitre 5

Classification de situations de conduite d'un 2RM

Sommaire

5.1	Introduction	50
5.1.1	État de l'art sur la classification de situations de conduite	50
5.1.2	Méthodologie de la classification de situations de conduite d'un 2RM	53
5.2	Modèle de mélange de Gaussiennes	53
5.2.1	Modèle de mélange	53
5.3	Les k-plus proches voisins (k-ppv)	58
5.4	Séparateurs à Vaste Marge	59
5.5	Forêts aléatoires	61
5.6	Approche adaptée aux données séquentielles	63
5.6.1	Généralités sur les Chaînes de Markov	63
5.6.2	Modèle de Markov caché discret	64
5.6.3	Modèle de Markov caché continu	68
5.7	Résultats de la classification de situations de conduite d'un 2RM	69
5.7.1	Critères d'évaluation des performances des classifieurs	69
5.7.2	Apprentissage des modèles	70
5.7.3	Sélection des hyper-paramètres des modèles	70
5.7.4	Évaluation des résultats et discussion	72
5.8	Conclusion	78

5.1 Introduction

Dans ce chapitre, nous présentons une des contributions de ces travaux de thèse qui concerne la classification de situations de conduite d'un 2RM. La première partie du chapitre est consacrée à l'état de l'art abordant la question de classification des situations de conduite que nous avons formalisé comme un problème de classification supervisée. Par la suite, nous évoquons les différentes étapes et algorithmes pouvant être envisagés pour mettre en place cette procédure de classification de situations de conduite d'un 2RM. Dans la dernière partie du chapitre, nous présenterons la mise en œuvre des différentes approches pour la reconnaissance de quatre situations de conduite (ligne droite, virage à gauche, virage à droite et rond-point) et les résultats de la sélection des capteurs les plus pertinents pour atteindre cet objectif.

5.1.1 État de l'art sur la classification de situations de conduite

Nous ne présenterons ici que les études basées sur l'apprentissage automatique qui nous ont paru les plus pertinentes pour la classification de situations de conduite de véhicule à quatre roues. En ce qui concerne les recherches effectuées sur les deux-roues motorisés, nous n'avons pas trouvé de publications consacrées spécifiquement à cette thématique de recherche. Cependant, l'analyse et la classification des situations de conduite pour les quatre roues a fait l'objet de plusieurs travaux de recherche dans la littérature, mais avec des visées applicatives différentes. Dans ce qui suit, nous allons citer les travaux les plus significatifs

Ainsi, Nechyba et Xu [39] ont appliqué les modèles de Markov cachés (HMMs) sur des données collectées à partir d'un simulateur de conduite pour comparer les stratégies de contrôle des conducteurs. Un modèle HMM discret a été associé à chaque conducteur à partir des indicateurs de comportement. Ces derniers sont obtenus par quantification vectorielle des actions de contrôle des conducteurs. La classification d'une nouvelle série de données se fait grâce à une mesure de similarité entre les HMMs préalablement construits et celui de la série à classer. Les auteurs ont montré que l'utilisation de cette mesure de similarité permet de comparer des trajectoires issues de plusieurs processus stochastiques à dimension vectorielle.

Dans une autre étude, Pentland et Liu [40] ont recueilli des données d'un simulateur de conduite (Nissan 240SX) telles que: l'angle du volant, la position du frein et la position de l'accélérateur. Des commandes textuelles présentées sur l'écran du simulateur sont demandées au conducteur pour les exécuter. Les commandes présentées étaient: arrêter, tourner à gauche ou à droite sur la prochaine intersection, changement de voie et dépassement de la voiture de devant. Le but de cette étude était de prédire les intentions du conducteur sur la base de ses premières actions. Les HMMs ont été utilisés pour reconnaître les situations de conduite 2s après la présentation de la commande

textuelle. Le taux de reconnaissance obtenu est de 95,24%, permettant aux auteurs de déduire que les intentions de conduite d'un conducteur peuvent être précisément reconnues très rapidement après le début de l'action.

Kari Torkkola [41] s'est inspiré des résultats obtenus dans le domaine de la reconnaissance automatique de la parole en utilisant un HMM hiérarchique. Il a ainsi modélisé une séquence de conduite comme une succession de manœuvres (situation de conduite). Chaque manœuvre est considérée comme une suite de sous-unités (actions) qu'il a appelé "drivemes". Pour la validation de la méthodologie proposée, une base de données issues d'un simulateur de conduite a été utilisée. Les données ont été recueillies sur quatre conducteurs sur une durée de 15 mn. Les situations de conduite les plus fréquentes rencontrées au cours de cette expérimentation sont : le changement de voie à gauche, le changement de voie à droite, le démarrage, l'arrêt, le zigzag. Les données de mesures des capteurs utilisés par l'auteur sont au nombre de 20. Ces capteurs sont les accéléromètres, les gyromètres, le frein, l'angle du volant, l'état du clignotant, la vitesse etc. A partir de ces 20 capteurs, un ensemble de 138 caractéristiques a été extrait. Un algorithme de sélection de caractéristiques de type Wrapper (Random Forests) a été mis en œuvre pour la sélection de 38 caractéristiques les plus pertinentes qui ont servi à l'apprentissage d'un modèle HMM continu. L'auteur a montré des résultats probants en terme de F-mesure, de rappel et de précision avec beaucoup de pertinence de sa méthodologie pour la reconnaissance des situations de conduite.

Lors de ses travaux sur l'établissement d'un cadre d'analyse des données recueillies sur les véhicules et la modélisation de leurs évolutions pour les mettre en correspondance avec les comportements de conduite, Dapzol [6] a étendu le modèle de Markov caché au modèle semi-Markovien caché pondéré'. Cela lui a permis d'intégrer des contraintes temporelles liées à l'activité de conduite et à la notion de pondération associée aux différentes variables. L'auteur a fait un recueil de données de 1200 séquences de conduite, à partir desquelles l'auteur a construit un catalogue de modèles de l'évolution des capteurs pour 36 situations de conduite. Grâce à des méthodes issues de la classification hiérarchique, il a pu déterminer dans quelle situation, ou dans quel groupe de situations, se trouve le conducteur à partir d'un enregistrement inconnu. Le taux de reconnaissance été de 75% pour une partition étendue (26 activités de conduite), 87% pour une partition intermédiaire (18 activités de conduite) et 90% pour une partition réduite (12 activités de conduite).

Dejan Mitrovic [42] quant à lui, a utilisé des données de mesures issues des capteurs installés sur un véhicule équipé d'accéléromètres, de gyroscopes et d'un récepteur GPS, pour la classification de situations de conduite (tourner à gauche, tourner à droite, virage à gauche, virage à droite, rond-point à gauche, rond-point à droite et rond-point tout droit). Pour des raisons de coût, l'auteur a utilisé seulement des accéléromètres et la vitesse fournie par le GPS pour la reconnaissance de ces situations de conduite

avec le modèle de Markov caché. Dans une perspective d'utiliser son système de reconnaissance de situations de conduite en temps réel, l'auteur a implémenté un HMM discret vu que le temps de calcul est plus faible que celui d'un HMM continu. Pour cela, une quantification vectorielle est utilisée pour traduire les données analogiques en symboles discrets. Pour l'apprentissage et la reconnaissance des situations de conduite, l'auteur a utilisé un HMM avec une topologie gauche-droite afin de mieux cerner les caractéristiques dynamiques des données. Un modèle HMM pour chaque situation de conduite à reconnaître a été construit. Pour la classification d'une nouvelle série de donnée, le modèle HMM qui maximise sa probabilité représente la situation de conduite recherchée. L'auteur a montré que son système arrive à reconnaître avec un taux de bonne classification de 98,3% des situations de conduite recherchées.

Toutefois, la seule étude dans la littérature liée à la classification de situations de conduite d'un autre moyen de déplacement est le travail de Thepvilojanapong. N, et al [43] sur les vélos. Les auteurs ont appliqué les HMMs sur des données collectées à partir d'un téléphone portable (Android phone Xperia) équipé d'un accéléromètre sur les trois axes (X-Y-Z), d'un magnétomètre sur les trois axes et d'un récepteur GPS) est monté sur un vélo. Leur objectif était de reconnaître quatre situations de conduite : ligne droite, virage à gauche/droite, arrêt et trajectoire sinusoïdale. Dans l'implémentation du modèle HMM, les auteurs ont utilisé seulement les accélérations et les forces magnétiques longitudinale et latérale. Le taux de reconnaissance était de 98%.

Ces différentes études sur la classification de situations de conduite pour un véhicule à quatre roues nous ont incité à poursuivre sur cette voie pour les 2RMs tout en tenant compte des particularités et spécificités de la conduite de ce mode de transport :

- Certains capteurs utilisés dans la plupart des véhicules à quatre roues ne peuvent pas être installés sur une moto sans engendrer de modification dans la dynamique de cette dernière.
- Dans la plupart des études présentées précédemment, les données utilisées pour la reconnaissance des situations de conduite sont issues des simulateurs de conduite. Ainsi, les résultats obtenus peuvent être biaisés du fait qu'il n'est pas certain que les actions de conduite soient les mêmes sur un simulateur qu'en situation réelle. D'où l'intérêt d'utiliser des données issues d'une expérimentation réelle.
- La complexité de la dynamique d'un 2RM par rapport à celle des véhicules à quatre roues rend l'activité de conduite d'un 2RM plus complexe à analyser.

5.1.2 Méthodologie de la classification de situations de conduite d'un 2RM

Afin d'atteindre notre objectif, nous avons adopté une méthodologie modulaire simple, basée sur une approche à base d'apprentissage statistique. Cette approche permet de proposer des solutions mieux adaptées à ce type de problème grâce à la prise en compte de l'aspect multi-variés des données et la corrélation mécanique qui existe entre les paramètres mesurés sur la dynamique du 2RM. La démarche que nous proposons pour la classification de situations de conduite d'un 2RM est une démarche classique de classification pour la reconnaissance des formes. A partir des données collectées, nous avons mis une stratégie qui se définit en trois niveaux de traitement :

1. une étape de pré-traitement et de labellisation des données où il s'agit essentiellement de mettre en forme les données;
2. une étape de mise en œuvre d'un ensemble de classifieurs;
3. une étape finale de classification et d'évaluation des méthodes utilisées.

La méthodologie proposée est présentée avec ses différentes étapes dans la figure 5.1.

La première partie du chapitre sera consacrée à la description des approches d'apprentissage utilisées à savoir: le modèle de mélange de gaussiennes, les k -plus proches voisins, les séparateurs à vaste marge, les forêts aléatoires et les modèles de Markov cachés. La dernière partie du chapitre sera consacrée à la mise en œuvre de ces différentes approches pour la reconnaissance de quatre situations de conduite ligne droite, virage à gauche virage à droite et rond-point. Dans ce qui suit, nous donnerons un aperçu des méthodes d'apprentissage automatique utilisées pour la classification de situations de conduite. Le problème de classification est considéré dans un contexte supervisé.

5.2 Modèle de mélange de Gaussiennes

5.2.1 Modèle de mélange

Le modèle de mélange a été largement développé, appliqué et utilisé dans la classification, le clustering, l'estimation de densités et la reconnaissance de formes, comme le montre [20, 44, 45]. En classification automatique, l'utilisation d'un modèle de mélange revient à supposer que les individus à classer sont issus d'un modèle de mélange dont chaque composante représente une classe.

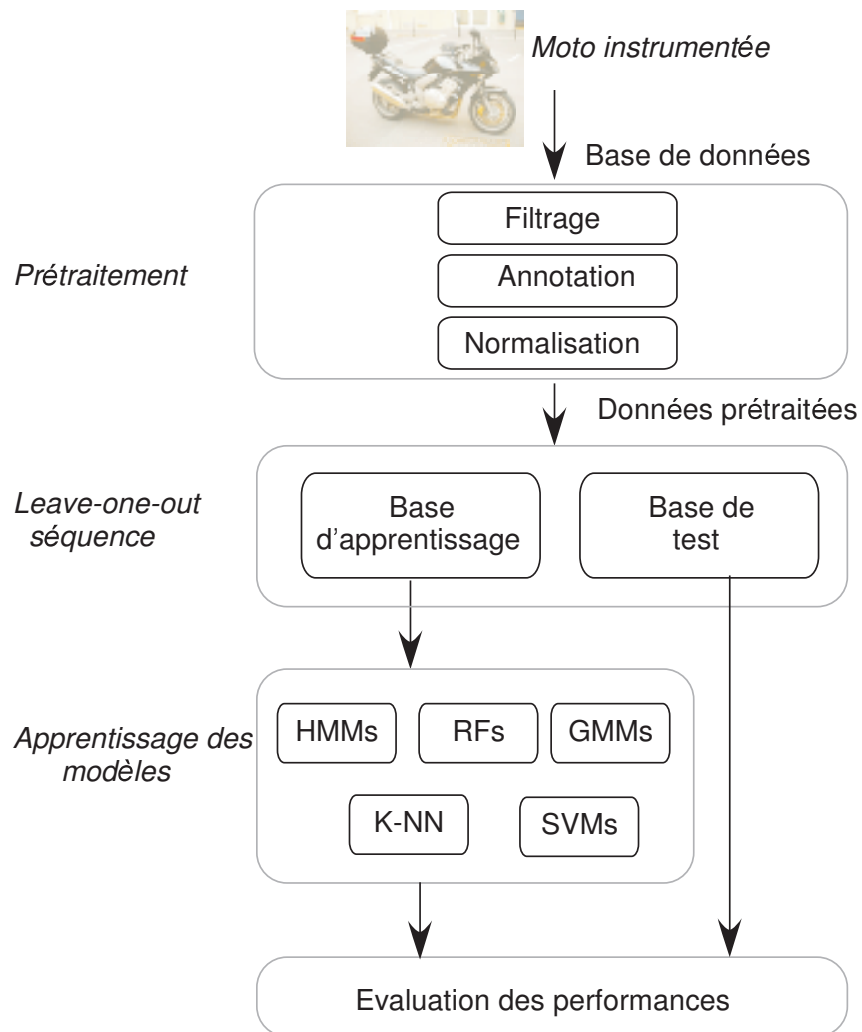


Figure 5.1: Synopsis de la méthodologie proposée.

Soit z une variable aléatoire discrète (binomiale ou multinomiale) qui prend ses valeurs dans l'ensemble fini $Z = \{1, \dots, K\}$ et $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_i, \dots, \mathbf{x}_N)$ un échantillon à N réalisations indépendantes. Le modèle de mélange modélise la densité de \mathbf{x} en une combinaison linéaire pondérée des densités de K composantes. Dans un cadre général, la densité du mélange de \mathbf{x} est donnée par:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{k=1}^K p(z = k) p(\mathbf{x} | z = k) \\ &= \sum_{k=1}^K \pi_k f_k(\mathbf{x}). \end{aligned} \tag{5.1}$$

Où

- K représente le nombre de composantes
- $\pi_k = p(z = k)$ représente la probabilité qu'un point de donnée sélectionné de façon aléatoire est généré par la composante k . Ces quantités sont généralement regroupées dans un vecteur $(\pi_1, \pi_2, \dots, \pi_k, \dots, \pi_K)$ appelé vecteur des proportions de mélange. Les éléments de ce vecteur sont positifs ($\pi_k \geq 0 \forall k$) et leur somme vaut 1 ($\sum_{k=1}^K \pi_k = 1$).
- $f_k(\mathbf{x})$ représente la densité de probabilité des composantes.

On suppose souvent que les composantes $f_k(\mathbf{x})$ du mélange sont paramétrées par α_k . Ainsi, la densité du mélange de \mathbf{x}_i peut s'écrire sous la forme:

$$f(\mathbf{x}_i; \boldsymbol{\theta}) = \sum_{k=1}^K \pi_k f_k(\mathbf{x}_i; \alpha_k). \tag{5.2}$$

$\boldsymbol{\theta} = (\pi_1, \dots, \pi_K, \alpha_1, \dots, \alpha_K)$ est le vecteur des paramètres du modèle de mélange qui appartient à l'espace des paramètres Θ .

L'estimation des paramètres d'un tel modèle peut se faire de différentes méthodes, les plus connues sont le maximum de vraisemblance [20] et les méthodes bayésiennes (Maximum A Posteriori (MAP)) [46, 47]. Dans ce qui suit, nous nous intéresserons à la méthode basée sur le maximum de vraisemblance et en particulier à l'algorithme le plus couramment utilisé, l'algorithme Espérance-Maximisation (EM) [48, 49].

L'objectif de cet algorithme est de maximiser la log-vraisemblance en fonction des paramètres du modèle $\boldsymbol{\theta} = (\pi_1, \dots, \pi_K, \alpha_1, \dots, \alpha_K)$. La log-vraisemblance des données

observées $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_i, \dots, \mathbf{x}_N)$ peut être écrite sous la forme suivante:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, \mathbf{X}) &= \log \prod_{i=1}^N p(\mathbf{x}_i, \boldsymbol{\theta}) \\ &= \sum_{i=1}^N \log \sum_{k=1}^K \pi_k f_k(\mathbf{x}_i; \boldsymbol{\alpha}_k).\end{aligned}\tag{5.3}$$

La maximisation de cette fonction selon $\boldsymbol{\theta}$ est très complexe. Cependant, cette fonction peut être maximisée en utilisant des procédures itératives telles que la méthode de Newton-Raphson ou la méthode du gradient. Dans ce qui suit, nous allons présenter l'algorithme Espérance-Maximisation (EM) pour le modèle de mélange paramétré puis son application à un type particulier de mélange qui est le mélange de Gaussiennes. Selon le problème à traiter, plusieurs modèles de mélange peuvent être envisagés, entre autres, le modèle de mélange de lois de Weibull pour l'étude de la durée de vie, le modèle de mélange de lois multinomiales pour la classification automatique, le modèle de mélange de gaussiennes etc. Dans le cadre de ces travaux de thèse, nous proposons le modèle de mélange de gaussiennes pour la classification de situation de conduite.

L'algorithme EM [48, 49] est une méthode itérative qui permet de trouver le maximum de vraisemblance dans le cadre des modèles qui dépendent des variables latentes non observées. Pour chaque itération, l'algorithme EM est composé de deux étapes, l'étape Espérance (E) et l'étape de Maximisation (M).

Dans le cas du modèle de mélange de gaussiennes, la forme de la loi conditionnelle utilisée quand les observations sont à valeurs dans \mathbb{R}^p est la loi normale multivariée donnée par:

$$\mathcal{N}(x_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{p}{2}} |\det(\boldsymbol{\Sigma}_k)|^{\frac{1}{2}}} \exp\left(\frac{-1}{2} (x_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (x_i - \boldsymbol{\mu}_k)\right).\tag{5.4}$$

Ainsi, le modèle de mélange de gaussiennes s'écrit:

$$f_k(x_i; \boldsymbol{\alpha}_k) = \sum_{k=1}^K \pi_k \mathcal{N}(x_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),\tag{5.5}$$

où $\boldsymbol{\alpha}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ contient le vecteur de moyennes $\boldsymbol{\mu}_k$ et la matrice de variance-covariance $\boldsymbol{\Sigma}_k$ de la $k^{\text{ème}}$ composante. La log-vraisemblance des données observées d'un tel modèle est donnée par:

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{X}) = \sum_{i=1}^N \log\left(\sum_{k=1}^K \pi_k \mathcal{N}(x_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right).\tag{5.6}$$

Cette log-vraisemblance est itérativement maximisée par l'algorithme EM [48]. La log-vraisemblance des données complétées dans ce cas est donnée par:

$$\mathcal{L}_c(\boldsymbol{\theta}; \mathbf{X}, \mathbf{z}) = \sum_{i=1}^N \sum_{k=1}^K z_{ik} \log(\pi_k \mathcal{N}(x_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)), \quad (5.7)$$

avec z_{ik} est un indicateur à valeur binaire qui vaut 1 quand la variable x_i est générée par la $k^{\text{ème}}$ composante du mélange, 0 sinon.

Ainsi, à partir des paramètres initiaux $\boldsymbol{\theta}^{(0)} = (\pi_1^{(0)}, \dots, \pi_K^{(0)}, \boldsymbol{\alpha}_1^{(0)}, \dots, \boldsymbol{\alpha}_K^{(0)})$, avec $\boldsymbol{\alpha}_k^{(0)} = (\boldsymbol{\mu}_k^{(0)}, \boldsymbol{\Sigma}_k^{(0)})$, l'algorithme EM alterne entre les deux étapes décrites ci-dessous jusqu'à convergence vers un maximum local de la fonction de la log-vraisemblance complétée.

Étape "E" (Espérance): cette étape consiste à calculer l'espérance de la log-vraisemblance des données complétées voir équation (5.8), en tenant compte des observations \mathbf{X} et de la valeur courante de $\boldsymbol{\theta}^{(q)}$.

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}) &= \mathbb{E}[\mathcal{L}_c(\boldsymbol{\theta}; \mathbf{X}, \mathbf{z}) | \mathbf{X}; \boldsymbol{\theta}^{(q)}] \\ &= \sum_{i=1}^N \sum_{k=1}^K \tau_{ik}^{(q)} \log(\pi_k \mathcal{N}(x_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)). \end{aligned} \quad (5.8)$$

Dans cette étape, on calcule donc les probabilités *a posteriori*

$$\tau_{ik}^{(q)} = p(z_{ik} = k | x_i; \boldsymbol{\theta}^{(q)}) = \frac{\pi_k \mathcal{N}(x_i; \boldsymbol{\mu}_k^q, \boldsymbol{\Sigma}_k^q)}{\sum_{l=1}^K \pi_l \mathcal{N}(x_i; \boldsymbol{\mu}_l^{(q)}, \boldsymbol{\Sigma}_l^{(q)})}. \quad (5.9)$$

que x_i provient de la densité de la $k^{\text{ème}}$ composante.

Étape "M" (Maximisation): Cette étape consiste à maximiser l'espérance de la log-vraisemblance des données complétées, en tenant compte des observations \mathbf{X} et de la valeur courante de $\boldsymbol{\theta}^{(q)}$. Dans le cas de mélange de gaussiennes, la maximisation de la fonction Q par rapport à $(\boldsymbol{\mu}_k$ et $\boldsymbol{\Sigma}_k)$ peut se faire selon les formules de mise à jour suivantes [20]:

$$\boldsymbol{\mu}_k^{(q+1)} = \frac{1}{n_k^{(q)}} \sum_{i=1}^N \tau_{ik}^{(q)} x_i, \quad (5.10)$$

$$\boldsymbol{\Sigma}_k^{(q+1)} = \frac{1}{n_k^{(q)}} \sum_{i=1}^N \tau_{ik}^{(q)} (x_i - \boldsymbol{\mu}_k^{(q+1)})(x_i - \boldsymbol{\mu}_k^{(q+1)})^t. \quad (5.11)$$

Quant à la maximisation de la fonction auxiliaire Q par rapport à π_k , il suffit d'utiliser la formule suivante:

$$\pi_k^{(q+1)} = \sum_{i=1}^N \frac{\tau_{ik}^{(q)}}{N} = \frac{n_k^{(q)}}{N}. \quad (5.12)$$

$n_k^{(q)}$ représente le nombre d'échantillons qui appartiennent à la composante k dans l'itération q .

Les deux étapes E et M sont alternées de façon itérative jusqu'à ce que la variation de la log-vraisemblance entre deux itérations successives soit inférieure à un certain seuil donné. Le pseudo-code 1 résume l'algorithme EM pour le modèle de mélange gaussien.

Algorithme 1 Pseudo-code de l'algorithme EM

Entrées: Ensemble de données \mathbf{X} et nombre de composantes K

```

1: Initialisation:
2:  $q \leftarrow 0$ ;
3: fixer un  $\epsilon > 0$ ;
4:  $\theta^{(0)} = (\pi_1^{(0)}, \dots, \pi_K^{(0)}, \alpha_1^{(0)}, \dots, \alpha_K^{(0)})$ , avec  $\alpha_k^{(0)} = (\mu_k^{(0)}, \Sigma_k^{(0)})$ ;
5: tantque  $\mathcal{L}_c^q > \epsilon$  faire
6:   Étape E
7:   pour  $k=1, \dots, K$  faire
8:     Calculer  $\tau_{ik}^{(q+1)}$  en utilisant l'équation (5.9)
9:   fin pour
10:  pour  $k=1, \dots, K$  faire
11:    Étape M
12:    Calculer  $\pi_k^{(q+1)}$  en utilisant l'équation (5.12)
13:    Calculer  $\mu_k^{(q+1)}$  en utilisant l'équation (5.10)
14:    Calculer  $\Sigma_k^{(q+1)}$  en utilisant l'équation (5.11)
15:  fin pour
16:   $q \leftarrow q + 1$ ;
17: fin tantque

```

Sorties:

$$\hat{\theta} = \theta^{(q)}$$

$$\widehat{\tau}_{ik} = \tau_{ik}^{(q)}.$$

5.3 Les k -plus proches voisins (k -ppv)

L'algorithme des k -plus proches voisins (ou k -nn de l'anglais k -nearest neighbors algorithm) [50], est un algorithme de classification connu pour sa simplicité et ses performances (voir [51]). Un article récent réalisé par [52] résume plusieurs améliorations de l'algorithme k -ppv. Cet algorithme fait partie de la famille des approches non-paramétriques dans la mesure où aucune hypothèse sur la répartition des différentes classes ou la nature des surfaces séparatrices est supposée connue. La phase d'apprentissage requière l'ensemble de la base étiquetée \mathcal{S} . La classification de nouveaux individus s'opère en a) calculant la distance entre cet individu avec tous les individus issus de la base d'apprentissage. Cette distance appelée également fonction de similarité peut être Euclidienne ou adaptative; b) puis en sélectionnant les k plus proches voisins

pour assigner comme étiquette de sortie la classe majoritaire de ces derniers (voir le pseudo-code l'algorithme k -ppv 2). Les performances de cette approche dépendent de la valeur de k qui est en général déterminée par une validation croisée.

Algorithme 2 Pseudo-code de l'algorithme k -ppv

Entrées: Ensemble de données d'apprentissage $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$

Nombre de plus proches voisins k

- 1: Une nouvelle instance \mathbf{x}' à classifier
- 2: **pour** $i=1, \dots, N$ **faire**
- 3: Calculer la distance $d(\mathbf{x}', \mathbf{x}_i)$ entre \mathbf{x}_i et \mathbf{x}'
- 4: **fin pour**
- 5: Sélectionner les k distances les plus proches de \mathbf{x}'
- 6: Affecter \mathbf{x}' à la classe majoritairement représentée par ses voisins

Sorties: \hat{y} est la classe assignée à \mathbf{x}' (classe majoritaire parmi les k plus proches voisins de \mathbf{x}'). .

5.4 Séparateurs à Vaste Marge

Les séparateurs à vaste marge¹ Les SVM ont été introduits par V.Vapnik en 1995 dans son ouvrage "*The nature of statistical learning theory*" [53]. Les SVM représentent depuis une dizaine d'années une approche largement reconnue dans la communauté de la classification supervisée pour ses excellentes performances globales se hissant généralement parmi les meilleurs. Depuis, différentes variantes de l'algorithme SVM ont été proposées par plusieurs chercheurs et appliquées à divers domaines.

Cette approche est basée sur la théorie de la Minimisation du Risque Empirique (MRE) et celle du Risque Structurel (MRS). Dans le cas où les données sont séparables, l'idée principale est de trouver un hyperplan (séparateur) $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$ qui maximise la distance entre les exemples positifs $y_i = +1$ et négatifs $y_i = -1$ et dont la séparation (ou marge : distance séparant la frontière du plus proche exemple) est aussi grande que possible (la marge entre les deux classes vaut $\frac{2}{\|\mathbf{w}\|}$). Comme le montre la figure (5.2), la partie gauche nous montre le cas où on a un hyperplan optimal avec une grande marge, ce qui nous permet de bien classer un nouvel exemple malgré le fait qu'il soit localisé dans la marge. Cependant, sur la partie droite malgré le fait que l'on ait trouvé un hyperplan optimal, l'exemple se voit mal classé, du fait que la marge est petite. Maximiser la marge revient à minimiser $\|\mathbf{w}\|$ ce qui équivaut à résoudre un problème d'optimisation quadratique sous contraintes.

Dans le cas des données non linéairement séparables (pas d'hyperplan séparant tous les exemples), on introduit des variables ξ_i de relaxation "ressort" pour assouplir

¹ *Support Vectors Machines* est littéralement traduit par certains auteurs par l'expression "Machines à Vecteurs de Support". Nous lui préférons "Séparateurs à Vaste Marge" qui nous semble plus adéquate.

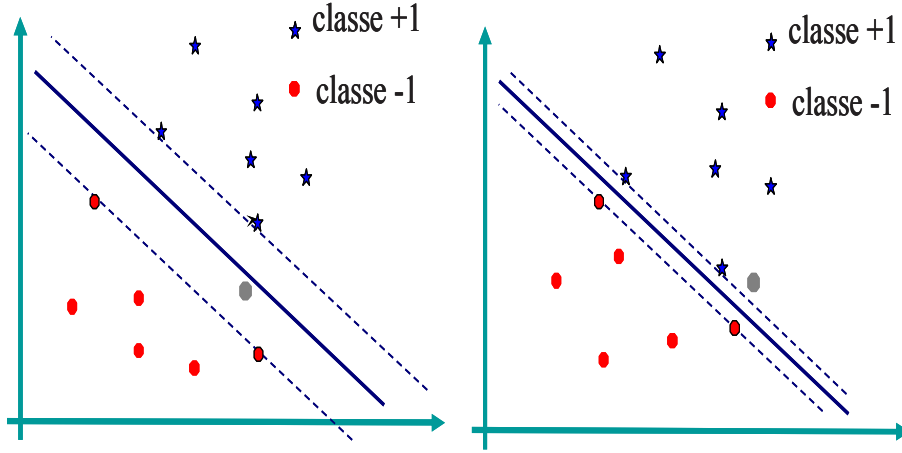


Figure 5.2: Illustration de l'impact de la maximisation de la marge sur la capacité de généralisation. Le point gris est un élément de l'ensemble de test.

les contraintes et contrôler au travers de la variable C . Sous sa forme primaire et en intégrant ces variables de relaxation, la fonctionnelle à optimiser s'écrit :

$$\begin{cases} \min_{\mathbf{w}, b} & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_i \xi_i \\ & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \forall i. \end{cases} \quad (5.13)$$

Ce problème peut se réécrire sous forme duale en introduisant des multiplicateurs de Lagrange pour chaque contrainte.

$$\begin{cases} \max & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \mathbf{x}_j) \\ & 0 \leq \alpha_i \leq C, \forall i \\ & \sum_{i=1}^N \alpha_i y_i = 0. \end{cases} \quad (5.14)$$

La fonction de décision associée s'exprime par : $f(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^N \alpha_i^* y_i (\mathbf{x}_i \mathbf{x}) + b \right)$ à partir des paramètres α^* et b solutions du problème précédent.

Vapnik propose de projeter les points dans un nouvel espace \mathcal{H} de Hilbert muni d'un produit scalaire $\langle \cdot, \cdot \rangle$ via une fonction $\varphi(\cdot)$. En pratique, cette transformation est implicite à travers le calcul du noyau de Mercer au travers de produits scalaires tels que $K(\mathbf{x}_1, \mathbf{x}_2) = \langle \varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2) \rangle$ [53].

Plusieurs noyaux peuvent être envisagés à savoir: noyau linéaire, noyau polynomial, noyau RBF (Radial Basis Function) ...

On en déduit le problème dual qui a la même forme que dans le cas séparable à

savoir :

$$\begin{cases} \max \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ 0 \leq \alpha_i \leq C, \forall i \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{cases} \quad (5.15)$$

et la frontière de décision s'exprime sous la forme : $f(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^N \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \right)$.

Dans notre cas, un C -SVM à noyau RBF sera utilisé pour la classification des situations de conduite.

La forme générique de ce noyau est présenté par [53] :

$$\begin{aligned} K(\mathbf{x}, \mathbf{z}) &= \exp \left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2} \right) \\ &= \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2), \end{aligned} \quad (5.16)$$

avec $\gamma = \frac{1}{2\sigma^2}$, où $\|\cdot\|$ désigne la norme Euclidienne. La fonction de projection induite par ce type de noyau permet de projeter un exemple sur une fonction gaussienne représentant la similarité de l'exemple avec tous les points de \mathcal{X} .

Le paramètre σ permet de régler la largeur de la gaussienne. En prenant un σ grand, la similarité d'un exemple par rapport à ceux qui l'entourent sera assez élevée, tandis qu'en prenant un σ qui tend vers 0, l'exemple ne sera similaire à aucun autre. En resserrant fortement la gaussienne, le classifieur peut apprendre n'importe quel ensemble d'apprentissage sans commettre d'erreur. En fait, σ peut être considéré comme un autre paramètre permettant de contrôler la capacité du classifieur.

Ainsi, les deux hyper-paramètres à ajuster sont la constante contrôlant le compromis entre une erreur empirique et la marge (C) et le paramètre γ .

5.5 Forêts aléatoires

Il est nécessaire d'introduire les arbres de décision avant d'évoquer les forêts aléatoires. Les arbres de décision sont des algorithmes de classification couramment utilisés en apprentissage automatique. L'utilisation courante de cette méthode repose essentiellement sur plusieurs facteurs: simplicité, efficacité, facilement interprétables et possède une capacité à capturer des relations non linéaires entre les entrées et les sorties de système.

Dnition 5.5.1. *Un arbre de décision est un classifieur de partitionnement récursif de l'espace des variables d'intérêts. Il est constitué de nœuds, de branches et de feuilles.*

Les nœuds forment un arbre enraciné, ce qui signifie qu'il est construit avec un nœud appelé "racine" qui n'a pas de branche entrante. Tous les autres nœuds ont une branche

entrante. Un nœud avec des branches sortantes est appelé un nœud interne. Les autres nœuds sont appelés feuilles (également appelés nœuds terminaux ou de décision).

Nous pouvons donner un exemple d'arbre de décision le plus souvent utilisé, il s'agit de l'algorithme CART (Classification And Regression Trees) qui inclue les arbres de classification et de régression [54]. Il est caractérisé par le fait qu'il construit des arbres binaires, à savoir que chaque nœud interne a exactement deux branches sortantes. Le critère de division de nœuds utilisé par cet algorithme est le critère de Gini [54]. Après avoir construit l'arbre, l'élagage est effectué. Puis, on supprime les branches les moins informatives au sens de l'erreur de classification. Une caractéristique importante de CART est sa capacité à générer des arbres de régression dont les feuilles prédisent un nombre réel et non une classe. Dans ce cas, CART cherche des partitions qui minimisent l'erreur quadratique de prédiction.

Les forêts aléatoires forment une famille de méthodes qui consistent à la construction d'un ensemble (ou une forêt) d'arbres de décision. L'idée sous-jacente a été proposée par Breiman [55] qui fut l'un des premiers à montrer que l'agrégation de plusieurs versions d'un estimateur dans un ensemble peut améliorer considérablement les performances du classifieur en termes de précision. L'auteur montre que le modèle moyen sur plusieurs échantillons bootstrap a une erreur de généralisation inférieure que celui appris sur un seul ensemble.

Dans un article publié en 2001 [56], Breiman combine la méthode du bagging qui est l'acronyme de "bootstrap aggregating" [55] et la sélection aléatoire de la variable de partitionnement de chaque nœud [57] pour donner naissance à une nouvelle méthode appelée forêts aléatoires ou RF (Random Forest) en anglais. L'auteur montre empiriquement que les forêts aléatoires donnent des résultats qui sont en concurrence avec les méthodes les plus puissantes de l'apprentissage automatique. Bien que les principes originaux sont dus à plusieurs chercheurs, Breiman est souvent cité comme le fondateur des forêts aléatoires.

Soit un ensemble de données d'apprentissage $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ avec $\mathbf{x}_i \in \mathbb{R}^p$ où p est le nombre d'attributs, $y_i \in \mathbf{Y}$ tel que $\mathbf{Y} = \{c_1, c_2, c_3, \dots, c_K\}$. L'algorithme des forêts aléatoires est une approche qui combine la technique du bagging et la randomisation du choix de l'attribut de partitionnement des nœuds dans la construction de l'arbre de décision. Le bagging tel qu'il est présenté par Breiman est une méthode d'ensemble qui utilise le rééchantillonnage pour générer une collection de classifieurs qui participent par la suite à un vote majoritaire. Le bagging consiste à construire T classifieurs à partir de plusieurs échantillons bootstrap $(\mathcal{S}_t)_{t=1, \dots, T}$, composés de N -uplets (x_i, y_i) tirés d'une manière aléatoire avec remise dans \mathcal{S} . Une collection de T classifieurs sera utilisée par la suite pour classifier une nouvelle réalisation via un vote majoritaire.

Algorithme 3 pseudo code de construction d'une Forêt aléatoire

Entrées: $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^N$ un ensemble d'apprentissage

 T nombre d'arbres.

 q nombre d'attributs.

1: **pour** $t = 1$ à T **faire**

2: Générer un échantillon Bootstrap S_t ;

3: Construire un arbre de décision de types CART sur l'échantillon Bootstrap S_t ;

4: A chaque noeud, on choisit la variable de partitionnement du noeud sur la base des q attributs tirés aléatoirement parmi les p attributs

5: **fin pour**
Sorties: $(f_t)_{t=1, \dots, T}$.

5.6 Approche adaptée aux données séquentielles

Les modèles de Markov cachés ou HMM (Hidden Markov Model) sont une classe de modèles de données latentes largement utilisés dans de nombreux domaines d'application. Un HMM est un modèle statistique approprié pour la modélisation des données séquentielles dans lesquelles les échantillons successifs ne sont plus supposés indépendants. Ils ont été introduits par Baum [58] dans les années 1960-1970. Depuis, plusieurs variantes de ces modèles ont été proposées, entre autres, l'algorithme Input-Output HMM (IOHMMs) introduit par [59], le HMM autoregressif [60, 61, 62, 63, 64], le Modèle semi-Markov caché (de l'anglais Hidden Semi-Markov Model) [65], ... D'autres extensions du modèle peuvent être trouvés dans [65]. Dans les deux paragraphes suivants, nous décrivons deux types de HMM (le discret et le continu).

5.6.1 Généralités sur les Chaînes de Markov

Avant de décrire le modèle HMM, nous allons commencer par donner quelques définitions classiques sur les processus stochastiques, l'ordre d'une chaîne de Markov et sur les chaînes de Markov observables.

1. Processus stochastique: pour décrire l'évolution d'un système composé de plusieurs variables aléatoires (V.A) indépendantes au cours du temps, on utilise une famille de V.A indexée par le temps ' t ' et on parle alors de processus stochastique. Les chaînes de Markov sont un type de processus stochastique qui vérifie :

$$p(z_0, z_1, \dots, z_t, \dots, z_N) > 0 \quad (5.17)$$

$$p(z_{N+1}|z_0, z_1, \dots, z_t, \dots, z_N) = P(z_{N+1}|z_N) \quad (5.18)$$

tel que :

$z_{N(N>0)}$ est une suite de V.A dans un ensemble fini ou dénombrable \mathbb{E} , et N un entier naturel.

Dnition 5.6.1. *Ordre de la chaîne de Markov: La suite $(z_0, z_1, \dots, z_t, \dots, z_N)$ est une chaîne de Markov d'ordre $k \geq 1$ si quelque soit $p > k$, $p(z_p|z_1, z_2, \dots, z_{p-1}) = p(z_p|z_{p-k}, \dots, z_{p-1})$.*

2. Chaîne de Markov observable: La chaîne de Markov est dite observable, si elle est définie par ([64]):

- Les états de la chaîne $\mathbf{S} = (s_1, s_2, \dots, s_K)$, K étant le nombre d'états (on parle également de l'alphabet de la chaîne).
- Les transitions de passer d'un état s_k à l'état s_l qui sont définies par les probabilités $P(s_k|s_l)$. Ces transitions sont regroupées dans \mathbf{A} , appelée matrice de transition définie par:

$$\mathbf{A} = \{\mathbf{a}_{kl}\} = P(s_k|s_l) \quad \text{avec} \quad 1 \leq k, l \leq K, \quad (5.19)$$

avec $\sum_{l=1}^K \mathbf{a}_{kl} = 1$

- Les probabilités initiales. En d'autres termes les probabilités de commencer par un état s_k notées par $P(s_k)$. Ces probabilités sont regroupées dans un vecteur appelé vecteur de probabilité initial noté $\boldsymbol{\pi}$ défini par :

$$\begin{aligned} \boldsymbol{\pi} &= \{\boldsymbol{\pi}_k\} \\ &= P(s_k) \quad \text{avec} \quad 1 \leq k \leq K, \end{aligned} \quad (5.20)$$

avec $\sum_{k=1}^K \boldsymbol{\pi}_k = 1$

Dnition 5.6.2. *Un modèle de Markov observable $\boldsymbol{\lambda}$ est défini par une matrice de transition \mathbf{A} et un vecteur d'initialisation $\boldsymbol{\pi}$ avec $\boldsymbol{\lambda} = \{\boldsymbol{\pi}, \mathbf{A}\}$*

5.6.2 Modèle de Markov caché discret

Dans le modèle de Markov caché, contrairement au modèle de Markov observable, les états $\mathbf{S} = \{s_1, s_2, \dots, s_K\}$, ne sont pas observables directement, par contre, ils émettent des observations $\mathbf{v} = \{v_1, v_2, \dots, v_M\}$ qui sont pondérées par leurs probabilités d'émission.

Soit $\mathbf{z} = (z_0, z_1, \dots, z_t, \dots, z_N)$ une chaîne de Markov cachée d'ordre 1. Considérant la séquence d'observations $\mathbf{X} = (x_0, x_1, \dots, x_t, \dots, x_N)$, tel que z_t prend ses valeurs dans l'ensemble d'état \mathbf{S} et x_t prend ses valeurs dans l'ensemble \mathbf{v} .

Le modèle de Markov caché est défini par:

- K , le nombre d'état de la séquence d'état $\mathbf{S} = \{s_1, s_2, \dots, s_K\}$.

- M , le nombre d'observations de la séquence d'observations $\mathbf{v} = \{v_1, v_2, \dots, v_M\}$.
- Les transitions d'un état $z_t = s_k$ à un autre état $z_{t+1} = s_l$, définies par $P(z_{t+1} = s_l | z_t = s_k)$. Ces transitions sont regroupées dans la matrice $\mathbf{A} = \{\mathbf{a}_{kl}\}$ appelée matrice de transition avec:

$$\mathbf{a}_{kl} = P(z_{t+1} = s_l | z_t = s_k) \quad \text{avec} \quad 1 \leq k, l \leq K. \quad (5.21)$$

- La probabilité que l'état s_l émet l'observation v_m à l'instant t est définie par la probabilité $P(x_t = v_m | z_t = s_l)$. Ces probabilités sont regroupées dans la matrice d'observation $\mathbf{B} = \{\mathbf{b}_{lm}\}$.

$$\mathbf{b}_{lm} = P(x_t = v_m | z_t = s_l) \quad \text{avec} \quad 1 \leq l \leq K, 1 \leq m \leq M. \quad (5.22)$$

- Les probabilités initiales. Ces probabilités sont regroupées dans un vecteur appelé vecteur de probabilité initial noté $\boldsymbol{\pi} = \{\pi_k\}$ défini par :

$$\pi_k = P(z_1 = s_k) \quad \text{avec} \quad 1 \leq k \leq K, \quad (5.23)$$

Pour des raisons de simplicité d'écritures mathématiques, on notera par la suite s_k , s_l par k , l et v_m par m .

En somme, le modèle de Markov caché est un processus doublement stochastique dont un est markovien non observable qui peut être observé par un autre processus observable.

Dans la partie suivante nous allons discuter des trois problèmes traités par les HMMs.

Problème de reconnaissance

Étant donné un HMM $\boldsymbol{\lambda} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ et une séquence d'observations $\mathbf{X} = \{x_1, x_2, \dots, x_N\}$, il s'agit de déterminer la probabilité $P(\mathbf{X} | \boldsymbol{\lambda})$ pour que \mathbf{X} soit générée par le modèle. Il est clair que la probabilité $P(\mathbf{X} | \boldsymbol{\lambda})$ que \mathbf{X} soit générée par $\boldsymbol{\lambda}$ est égale à la somme des probabilités que \mathbf{X} soit générée en suivant tous les chemins pouvant l'émettre. Cette probabilité se calcule par l'algorithme Forward-Backward [64].

Problème d'analyse

Le problème posé consiste à identifier la séquence d'état $\mathbf{z} = \{z_1, z_2, \dots, z_N\}$ qui maximise la probabilité $P(\mathbf{X} | \boldsymbol{\lambda})$. En d'autres termes, étant donné un HMM $\boldsymbol{\lambda}$ et une séquence d'observations \mathbf{X} , l'objectif est de trouver la séquence d'états qui maximise la

probabilité d'avoir généré \mathbf{X} . Cette probabilité se calcule par l'algorithme de Viterbi [66].

Problème d'apprentissage

Étant donné une séquence d'observations \mathbf{X} , l'objectif est d'estimer les paramètres du modèle $\lambda = \{\mathbf{A}, \mathbf{B}, \pi\}$ afin de maximiser la probabilité $p(\mathbf{X}|\lambda)$. On sait que la distribution de la séquence d'état caché $\mathbf{z} = (z_1, \dots, z_N)$ dans le cas d'une chaîne de Markov cachée d'ordre 1 peut s'écrire de la façon suivante:

$$p(\mathbf{z}; \pi, \mathbf{A}) = p(z_1; \pi) \prod_{t=2}^N p(z_t | z_{t-1}; \mathbf{A}) \quad (5.24)$$

La distribution conditionnelle de la séquence d'observations $\mathbf{X} = (x_1, x_2, \dots, x_N)$ est donnée par:

$$p(\mathbf{X}|\mathbf{z}; \lambda) = p(x_1|z_1; \lambda) \prod_{t=2}^N p(x_t|z_t; \lambda) \quad (5.25)$$

Ainsi, nous obtenons la distribution conjointe suivante (la probabilité des données complètes):

$$\begin{aligned} p(\mathbf{X}, \mathbf{z}; \lambda) &= p(\mathbf{z}, \pi, \mathbf{A}) p(\mathbf{X}|\mathbf{z}; \lambda) \\ &= p(z_1; \pi) p(x_1|z_1; \lambda) \prod_{t=2}^N p(z_t|z_{t-1}; \mathbf{A}) p(x_t|z_t; \lambda) \end{aligned} \quad (5.26)$$

Soit $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$ le vecteur de paramètres du modèle à estimer. L'estimation de ces paramètres peut se faire par la méthode du maximum de vraisemblance donnée par l'équation suivante:

$$\begin{aligned} \mathcal{L}(\lambda; \mathbf{X}) &= \log p(\mathbf{X}; \lambda) = \log \sum_{\mathbf{z}} p(\mathbf{X}, \mathbf{z}; \lambda) \\ &= \log \sum_{\mathbf{z}} p(z_1; \pi) p(x_1|z_1; \lambda) \prod_{t=2}^N p(z_t|z_{t-1}; \mathbf{A}) p(x_t|z_t; \lambda) \end{aligned} \quad (5.27)$$

La maximisation de cette log-vraisemblance peut s'avérer difficile analytiquement, d'où l'utilisation de l'algorithme Baum Welch [67] qui est une version de l'algorithme EM appliqué aux HMMs.

Les formules de mise à jour sont définies comme suit [64]:

- mise à jour des nouvelles probabilités d'émission :

$$b_{km}^{(q+1)} = \frac{\sum_{t=1}^n \tau_{tk}^{(q)}}{\sum_{t=1}^N \tau_{tk}^{(q)}} \quad (5.28)$$

- mise à jour des nouvelles probabilités de transition:

$$a_{lk}^{(q+1)} = \frac{\sum_{t=2}^N \xi_t^{(q)}(l, k)}{\sum_{t=2}^N \tau_{tk}^{(q)}} \quad (5.29)$$

- mise à jour des nouvelles probabilités initiales:

$$\pi_k^{(q+1)} = \tau_{1k}^{(q)} \quad (5.30)$$

avec τ_{tk} la probabilité *a posteriori* de l'état k à l'instant t connaissant la séquence d'observations \mathbf{X} et les paramètres courants.

$\xi_t^{(q)}(l, k)$ représente la probabilité jointe de l'état k à l'instant t et l'état l à l'instant $t - 1$ connaissant la séquence d'observations \mathbf{X} et les paramètres courants $\boldsymbol{\lambda}^{(q)}$.

Différents types des HMMs

Les différents types des HMMs peuvent être classés en trois topologies à savoir :

Le modèle de Markov ergodique, le modèle de Markov Bakis et le modèle de Markov gauche-droite . Nous allons présenter le modèle de Markov gauche-droite qui a été utilisé dans ces travaux de thèse pour l'apprentissage des modèles HMMs.

Le modèle de Markov gauche-droite: la topologie gauche-droite est un cas particulier de la topologie Bakis dans laquelle la transition d'un état vers un autre état ne se fait que de l'état à indice inférieur vers les deux états à indice supérieur que ce dernier . Cette topologie est représentée sur la figure 5.3.

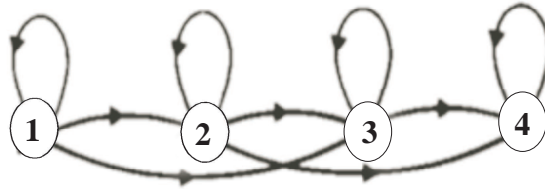


Figure 5.3: Illustration du modèle HMM gauche-droite.

5.6.3 Modèle de Markov caché continu

Dans ce qui a été présenté précédemment, nous n'avons considéré que le cas des observations discrètes. En d'autres termes, chaque observation x_t est un symbole de l'ensemble $v = \{v_1, v_2, \dots, v_M\}$. Dans certaines applications, il est nécessaire de considérer les observations continues. Certaines restrictions doivent alors être établies sur la forme de la densité de probabilité afin de garantir l'estimation des paramètres de cette dernière. La forme la plus courante dont les procédures de mise à jour des paramètres ont été formulées est celle des mélanges [64]:

$$b_k(\mathbf{X}) = \sum_{m=1}^M c_{km} \mathfrak{R}[\mathbf{X}, \boldsymbol{\mu}_{km}, \boldsymbol{\Sigma}_{km}] \quad (5.31)$$

Avec

- c_{km} sont les coefficients de pondération.
- \mathfrak{R} est une densité de probabilité (souvent, on utilise une densité Gaussienne)
- $\boldsymbol{\mu}_{km}$ les vecteurs de moyenne.
- $\boldsymbol{\Sigma}_{km}$ les matrices de covariance.

Ainsi pour l'estimation des paramètres du modèle, contrairement à ce qui a été présenté dans le cas discret, au delà des paramètres $(\boldsymbol{\pi}, \mathbf{A}$ et $\mathbf{B})$, les coefficients de pondération c_{km} , les vecteurs moyennes $\boldsymbol{\mu}_{km}$ et les matrices de covariance $\boldsymbol{\Sigma}_{km}$ sont à mettre à jour. La figure 5.4 donne une représentation graphique d'un modèle HMM continu.

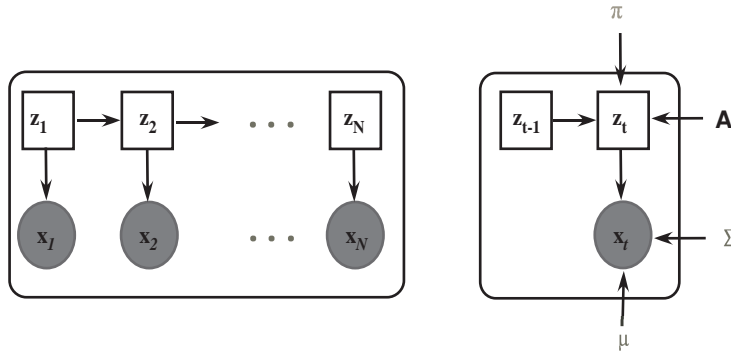


Figure 5.4: Modèle graphique de génération des données d'un modèle HMM continu.

Les formules de mise à jour sont définies comme suit [64]:

$$c_{km}^{(q+1)} = \frac{\sum_{t=1}^N \tau_t^{(q)}(km)}{\sum_{t=1}^N \sum_{m=1}^M \tau_t^{(q)}(km)} \quad (5.32)$$

$$\boldsymbol{\mu}_{km}^{(q+1)} = \frac{\sum_{t=1}^N \boldsymbol{\tau}_t^{(q)}(km) x_t}{\sum_{t=1}^N \boldsymbol{\tau}_t^{(q)}(km)} \quad (5.33)$$

$$\boldsymbol{\Sigma}_{km}^{(q+1)} = \frac{\sum_{t=1}^N \boldsymbol{\tau}_t^{(q)}(km) (x_t - \boldsymbol{\mu}_{km})(x_t - \boldsymbol{\mu}_{km})^T}{\sum_{t=1}^N \boldsymbol{\tau}_t^{(q)}(km)} \quad (5.34)$$

Avec $\boldsymbol{\tau}_t^{(q)}(km)$ est la généralisation de $\boldsymbol{\tau}_{tk}^{(q)}$ dans le cas discret au cas d'un simple mélange :

$$\boldsymbol{\tau}_t(km) = \frac{\boldsymbol{\alpha}_t(k) \boldsymbol{\beta}_t(k)}{\sum_{k=1}^K \boldsymbol{\alpha}_t(k) \boldsymbol{\beta}_t(k)} \frac{c_{km} \Re[\mathbf{X}, \boldsymbol{\mu}_{km}, \boldsymbol{\Sigma}_{km}]}{\sum_{o=1}^M c_{ko} \Re[\mathbf{X}, \boldsymbol{\mu}_{ko}, \boldsymbol{\Sigma}_{ko}]} \quad (5.35)$$

5.7 Résultats de la classification de situations de conduite d'un 2RM

Dans cette partie du chapitre, nous présentons et analysons les résultats de classification des différentes approches supervisées décrites précédemment. Rappelons que quatre situations de conduites ont été considérées : Ligne Droite (LD), Virage à Gauche (VG), Virage à Droite (VD) et Rond-Point (RP). Pour la reconnaissance des Arrêts (AR), un simple seuillage de la vitesse suffit pour la détection. La reconnaissance de ces situations de conduite est réalisée par un apprentissage supervisé. Pour cela, nous avons proposé de construire des modèles pour chaque situation de conduite à reconnaître puis les comparer avec de nouvelles séquences inconnues.

5.7.1 Critères d'évaluation des performances des classifieurs

Pour évaluer les performances des différentes approches de classification, nous avons utilisé le taux de bonne classification. Dans notre cas, les classes ne sont pas équiréparties (voir figure 3.10). Par exemple, le nombre d'échantillons de la situation de conduite VG est très inférieur à celui de la situation de conduite LD. Le taux de reconnaissance peut donc fournir une indication biaisée sur les performances du classifieur. Pour pallier ce problème, nous avons utilisé d'autres critères comme la moyenne du taux de reconnaissance (R) et l'écart type (Std), le rappel, la précision, la F-mesure, la F-mesure par classe et l'indice Kappa (κ_c). Comme indiqué dans le chapitre 4, notre base de données est constituée de 29 séquences de conduite dans laquelle nous avons 8 séquences de conduite avec "chute", 12 séquences de conduite avec "presque chute" et 11 séquences de conduite "normale". Dans cette partie de la thèse, nous allons utiliser uniquement les séquences de conduite normale. Ainsi pour évaluer les performances des différents classifieurs, nous avons utilisé la procédure de validation Leave One-Out Sequence [68]. En d'autres termes, le test s'effectue sur une séquence et l'apprentissage

s'effectue sur les séquences restantes. Cette procédure est répétée 11 fois (c-à-d pour chaque séquence).

5.7.2 Apprentissage des modèles

La reconnaissance des situations de conduite a été réalisée dans un contexte d'apprentissage supervisé à partir de données recueillies par une centrale inertielle (accéléromètres / gyromètres). Le vecteur d'état du système observé est représenté par: $\mathbf{x}^l = \{a_x, a_y, a_z, r_x, r_y, r_z\}$. Pour ce faire, nous avons choisi de construire un modèle pour chaque situation de conduite. Conduisant ainsi, à la construction de quatre modèles pour chaque approche, dédiés à reconnaître les situations de conduite: LD, VG, VD et RP. Chaque approche de classification dispose d'un ou plusieurs paramètres appelés hyper-paramètres contrôlant la qualité de prédiction du classifieur. Le choix de ces hyper-paramètres est une tâche cruciale. Dans le choix des hyper-paramètres du modèle, un compromis entre un modèle simple et un modèle complexe doit être trouvé, tout en tenant compte de la capacité de généralisation du modèle. Pour cela, nous avons opté pour une validation croisée avec la méthode de grille que nous discuterons ci-dessous.

5.7.3 Sélection des hyper-paramètres des modèles

La sélection automatique des paramètres des modèles peut être formulé comme un problème d'optimisation d'une fonctionnelle de coût f_{θ} (qui peut être le risque associé au modèle, une borne de l'erreur "Leave One Out", ...), avec θ étant les paramètres du modèle. Nous nous sommes basés sur la méthode de sélection par grille. Cette méthode consiste à fixer une grille de valeurs possibles (grille 2D ou 3D selon le nombre de hyper-paramètres à estimer). Par exemple, dans le cas d'un algorithme à deux hyper-paramètres C et γ , on fait varier le paramètre C dans un intervalle fixé $[C_{min}, C_{max}]$ avec des incréments ΔC et γ dans un intervalle $[\gamma^{min}, \gamma^{max}]$ avec des incréments $\Delta \gamma$. Alors, pour chaque couple de paramètres $\{C, \gamma\}$, on estime le risque du modèle associé par la méthode de validation leave out séquences. Ensuite, on sélectionne le jeu de paramètres $\{C^*, \gamma^*\}$ ayant obtenu le meilleur score (score minimum). Certes, cette méthode est coûteuse en temps de calcul, mais elle présente l'avantage de sélectionner les paramètres optimaux dans les intervalles choisis.

- **GMMs:** Comme indiqué précédemment le nombre de classe recherchées est 4, par conséquent le nombre de gaussiennes du modèle GMM est 4. Ici on cherche juste le type de gaussienne (diagonale, pleine ou sphérique). Et pour ce faire, Pour chaque type de gaussienne nous avons effectué la procédure du *leave one out sequence*. Nous avons retenu les gaussiennes diagonales.

- **k-NN**: Dans l'algorithme **k-NN** le seul paramètre à ajuster est le nombre de k plus proches voisins. Comme ce paramètre dépend essentiellement des données, nous avons fait varier k de 1 à 20, et pour chaque k nous avons estimé le risque du modèle associé. Le meilleur k que nous avons retenu pour ce cas d'étude est $k=10$.
- **SVMs**: Dans le cas des SVM, nous avons utilisé le C -SVM à noyau RBF: les deux hyper-paramètres à régler sont C et γ . Pour trouver les valeurs optimales de ces deux paramètres, nous avons utilisé la méthode de sélection par grille décrite ci-dessus. Pour cela, nous avons fait varier C de 1 à 10 et γ de -10 à 10. Les valeurs optimales obtenues pour l'algorithme C -SVM à noyau RBF sont: $(\log(C^*), \log(\gamma^*)) = (5, -9)$.
- **RFs**: Dans le cas des forêts aléatoires, le seul hyper-paramètre à régler est le nombre de forêts h que nous avons choisi par validation. Pour cela, nous avons fait varier le h de 1 à 100. La valeur optimale obtenue dans ce cas est $h = 50$.
- **CHMMs**: Comme décrit précédemment, il existe différentes topologies de HMM, selon les restrictions imposées sur la matrice de transition A . Il a été montré que le modèle de gauche-droite est l'un des meilleurs modèles dans la capture des caractéristiques dynamiques des données [42], en imposant un ordre temporel pour le modèle. Cette topologie a été largement utilisée dans un certain nombre de d'applications de reconnaissance de formes où l'aspect temporel intervient comme la reconnaissance vocale, la reconnaissance gestuelle, la reconnaissance de situations de conduite des conducteur de quatre roues. Comme le problème de la reconnaissance de l'activité de conduite d'un deux roues motorisé est similaire aux domaines mentionnés ci-dessus, nous nous sommes limités à l'utilisation des modèles gauche-droite. Une fois la topologie sélectionnée, il reste à régler les hyper-paramètres du modèle qui sont, le nombre d'états et le nombre de gaussiennes. Toujours en se fondant sur la méthode de grille en faisant varier le nombre d'états de 1 à 10 et le nombre de gaussiennes de 1 à 5, les hyper-paramètres optimaux obtenus sont 7 états pour les LD, 5 états pour les VG, 6 états pour les VD et 6 états pour les RP, chaque état est modélisé par un mélange de deux gaussiennes.
- **DHMMs**: Dans le cas des HMM discrets, nous avons besoin de convertir les données continues en symboles. Pour cela, nous avons utilisé la méthode de quantification vectorielle à l'aide de l'algorithme de clustering itératif k -means. Comme l'algorithme k -means est connu pour converger vers le minimum local, nous le répétons plusieurs fois avec différentes valeurs aléatoires pour les vecteurs de code initial afin de trouver la meilleure solution globale possible. Le k -means prend en entrée le vecteur de données \mathbf{x}^l et en sortie on aura le dictionnaire

D^l composé par les centroïdes de chaque signal ainsi que le vecteur de symboles associé. Ainsi dans les cas des HMMs discrets, les paramètres à régler sont le nombre de codes du dictionnaire et le nombre d'états. Nous avons retenu la même procédure que précédemment et nous avons choisi la topologie gauche-droite. Le choix du nombre de codes du dictionnaire est établi en faisant varier ce nombre de 1 à 200 et le nombre d'états de 1 à 10. Les hyper-paramètres optimaux obtenus sont 6 états pour les LD, 5 états pour les VG, 6 états pour les VD et 6 états pour les RP avec un nombre de codes de 128.

5.7.4 Évaluation des résultats et discussion

Les résultats obtenus avec les différentes méthodes de classification sont donnés dans le tableau 5.1. Ces résultats montrent que les taux de bonne classification dans la majorité des séquences est supérieur à 80%, sauf pour les séquences 6 et 7. Les taux de bonne classification obtenus avec les différentes méthodes de classification pour ces deux séquences est faible par rapport aux résultats obtenus pour les autres séquences. Afin de comprendre cette différence, nous avons entrepris une analyse plus fine des séquences 6 et 7. L'analyse vidéo montre que ces deux séquences ont été effectuées par le même conducteur. Nous avons constaté aussi un comportement erratique de ce conducteur dans certaines situations (par exemple, évitement d'une zone de travaux en roulant sur le trottoir et un manque de respect du code de la route). Pour garder une certaine homogénéité des conditions de mesures, ce participant est exclu de la présente étude. Désormais, le nombre total de séquences est $\mathcal{L} = 9$.

Table 5.1: Le taux de bonne classification des différents algorithmes d'apprentissage pour chaque séquence (le meilleur résultat est marqué en gras).

	1	2	3	4	5	6	7	8	9	10	11
GMMs (%)	80.7	95.8	80.9	82.7	78.9	62.8	74.3	70	84	86.3	88
k-NN (%)	84.6	95.8	76.1	86.2	78.9	60	61.5	76.6	84	77.2	92
SVMs (%)	84.6	95.8	80.9	82.7	78.9	60.	69.2	70	84	81.8	96
RFs (%)	88.4	95.8	80.9	86.2	78.94	65.7	64.1	80	84	81.8	88
DHMMs (%)	88.4	91.6	80.9	86.2	94.7	65.6	69.6	80	84	81.8	84
CHMMs (%)	88.4	87.5	90.4	82.7	89.4	71.4	74.3	76.6	88	90.9	84

Les taux de bonne classification globaux obtenus avec les différentes techniques de classification ainsi que leurs écarts-types sont résumés dans le tableau 5.2. On note que tous les taux globaux de bonne classification des classifieurs sont supérieurs à 82%. Les CHMMs donnent le taux de bonne classification global le plus élevé avec 86,6%, suivis par les DHMMs avec 85,7%, les forêts aléatoire avec 84,7%, les SVM avec 83,9%, le

5.7 Résultats de la classification de situations de conduite d'un 2RM

k-NN quant à lui, donne un taux de bonne classification global de 82,4% et les GMMs donnent un taux de 82,1%. On note aussi que l'algorithme des CHMMs est le meilleur en termes de taux de bonne classification global avec le plus faible écart-type et la plus grande valeur de F_1 -mesure, précision, rappel ainsi que le (κ_c) avec respectivement 82,6 %, 87,6 %, 78,1 % et 0,8108, suivis par DHMMs avec une F_1 -mesure, précision, rappel et (κ_c) 79,0 %, 79,3 % 78,7 % et 0.7805, respectivement.

En analysant les valeurs du κ_c on note qu'il y a un accord presque parfait ($\kappa_c \geq 0.81$) entre les résultats obtenus par les CHMMs et les classes réelles. Concernant les autres classifieurs on note qu'il y a un accord fort entre les résultats obtenus par ces derniers et les classes réelles ($\kappa_c \in [0,61, 0,80]$) (voir tableau 4.2 du chapitre 4).

Table 5.2: Les résultats obtenus en terme de F_1 -mesure par classe, F_1 -mesure global, κ_c et le taux de bonne classification global (R) et son écart type(Std) pour chaque algorithme (le meilleur résultat est marqué en gras).

	F_1 -mesure par classes (%)				F_1 -mesure (%)	κ_c	TR (%)
	LD	VG	VD	RP			
GMMs	89.1	50	83.3	70.2	70.2	0.7388	82.1 \pm 7.8
k-NN	89.1	29.4	86.2	73.1	75.8	0.7483	82.4 \pm 7.4
SVMs	89.2	47.3	85.9	75.6	78.4	0.7533	83.9 \pm 9.8
RFs	90.1	45	88.2	78.9	78.3	0.7652	84.7 \pm 7.6
DHMMs	91.2	65.3	87.9	68.4	79	0.7805	85.7 \pm 5
CHMMs	90.8	75	89.2	68	82.6	0.8108	86.4 \pm 4.5

Table 5.3: Comparaison statistique des différents classifieurs via le test de McNemar Z. Les flèches (\leftarrow, \uparrow) indiquent l'algorithme le plus performant.

	k-NN	SVMs	RFs	DHMMs	CHMMs
GMMs	\leftarrow 0.1905	0	\uparrow 0.4500	\uparrow 2.5600	\uparrow 1.6131
k-NN		\uparrow 0.0500	0	\uparrow 1.7857	\uparrow 1.8947
SVMs			\uparrow 0.3077	\uparrow 2.7222	\uparrow 1.6410
RFs				\uparrow 1.4545	\uparrow 1.6320
DHMMs					0

Comme présenté précédemment, nous avons utilisé le test de McNemar afin de savoir si la différence entre les performances des classifieurs est statistiquement significative. Les résultats du test de McNemar sont présentés dans le tableau 5.3. On constate une différence significative en faveur des HMMs comparés aux autres classifieurs (RFs, k-NN, SVMs et GMMs) pour une p-value de 5% ($Z > 1.6$). On note aussi qu'il n'y a pas

de différence significative entre les CHMMs et les DHMMs ($Z = 0$). On peut voir aussi une légère différence entre les RFs et les DHMMs ($Z = 1.4545$) en faveur des DHMMs, cette différence n'est pas significative vu que $Z < 1.6$. Concernant la différence entre les SVMs, the GMMs, the RFs and k -NN, cette différence n'est pas significative, c-à-d le test de McNemar tend vers zéro.

Résultats de la sélection des capteurs les plus pertinents

Nous avons montré dans la section précédente que les meilleurs résultats sont obtenus en utilisant les CHMMs. Raison pour laquelle nous choisissons ce modèle pour sélectionner la meilleure combinaison de capteurs afin d'améliorer les résultats de classification. Ici nous cherchons à sélectionner la meilleure combinaison de capteurs parmi les six capteurs $\{a_x, a_y, a_z, r_x, r_y, r_z\}$. L'approche que nous avons utilisée pour la sélection de capteurs les plus pertinents est réalisée via deux étapes (voir pseudo code de l'algorithme 4) comme suit:

- Génération des sous-ensembles de combinaisons de capteurs à l'aide d'une stratégie de recherche complète. Cette stratégie de recherche est exhaustive, elle permet de générer tous les sous-ensembles de combinaison de capteurs possible. Par conséquent, on assure la sélection du sous-ensemble optimal. Ainsi, 2^6 sous-ensembles de capteurs sont générés à partir des six capteurs $\{a_x, a_y, a_z, r_x, r_y, r_z\}$.
- Évaluation de tous les sous-ensembles générés selon la méthode wrapper pour sélectionner le meilleur sous-ensemble. Le processus de sélection des meilleurs sous-ensembles est un processus itératif. Dans chaque itération, les performances de chaque sous-ensemble sont évaluées. Ensuite, un sous-ensemble est sélectionné comme pertinent si le taux de bonne classification obtenu est supérieur à un seuil prédéfini; le seuil utilisé dans notre cas est le taux de bonne classification obtenu précédemment pour l'algorithme CHMMs à l'aide de six capteurs.

Parmi les 2^6 sous-ensembles générés neuf sous-ensembles ont été sélectionnés:

- $SE_1 = \{a_z, r_x, r_z\}$
- $SE_2 = \{r_x, r_y, r_z\}$
- $SE_3 = \{a_x, a_z, r_x, r_z\}$
- $SE_4 = \{a_x, r_x, r_y, r_z\}$
- $SE_5 = \{a_y, r_x, r_y, r_z\}$
- $SE_6 = \{a_z, r_x, r_y, r_z\}$

Algorithme 4 Pseudo code de la procédure de sélection de capteurs.

Entrées: Ensemble de capteurs $F = \{a_x, a_y, a_z, r_x, r_y, r_z\}$

- 1: **Initialisation:** Estimation du taux moyen de bonne classification R_0 obtenu avec six les capteurs
- 2: **Première étape: Recherche Exhaustive** génération de 2^6 sous-ensemble de capteurs F'_i avec $i = 1, \dots, 2^6$
- 3: **Deuxième étape: Évaluation et sélection**
- 4: **pour** $i = 1, \dots, 2^6$ **faire**
- 5: Apprentissage des modèles en utilisant le sous-ensemble F'_i
- 6: Estimation du taux de bonne classification R_i sous-ensemble F'_i
- 7: **si** $R_i > R_0$ **alors**
- 8: Le sous-ensemble F'_i est sélectionné
- 9: **sinon**
- 10: Le sous-ensemble F'_i est rejeté
- 11: **fin**
- 12: **fin pour**

Sorties: Sous-ensembles sélectionnés.

- $SE_7 = \{a_x, a_y, a_z, r_x, r_z\}$
- $SE_8 = \{a_x, a_y, r_x, r_y, r_z\}$
- $SE_9 = \{a_y, a_z, r_x, r_y, r_z\}$

Le tableau 5.4 représente le taux de bonne classification global des neufs sous-ensembles sélectionnés en utilisant le modèle des CHMMs. Ces résultats montrent que tous les taux de bonne classification sont supérieurs à 86,4% (taux de bonne classification obtenu avec les six capteurs). On note aussi que le taux de bonne classification le plus élevé (90,4%) est obtenu en utilisant SE_8 .

Table 5.4: Le taux de bonne classification obtenu pour les différents sous-ensembles sélectionnés en utilisant le modèle CHMM

	SE_1	SE_2	SE_3	SE_4	SE_5	SE_6	SE_7	SE_8	SE_9
CHMMs (%)	88.3	88.9	87	87.9	88.5	86.6	86.8	90.4	87.8

En analysant les résultats de la sélection de sous-ensembles de capteurs les plus pertinents, on constate la présence de la vitesse de lacet (r_z) et la vitesse de roulis (r_x) dans tous les sous-ensembles sélectionnés. Ces variables sont des mesures importantes pour décrire l'activité de conduite. Ces deux capteurs mesurent le comportement dynamique de la moto induit par les actions du conducteur sur cette dernière. Le capteur r_x est la conséquence de l'action du conducteur sur la moto effectuée soit par son corps, soit par l'intermédiaire du guidon pour garder la moto en équilibre relatif dans les virages. Le capteur r_z mesure le taux de changement de l'angle d'orientation de la moto,

qui est aussi une conséquence directe de l'action du conducteur sur le guidon. Ces actions traduisent les intentions du conducteur (pour aller tout droit, tourner à gauche ou tourner à droite). On constate aussi que les meilleurs résultats sont obtenus en utilisant sous-ensemble SE_8 , qui est composé de tous les capteurs sauf a_z . Ce résultat peut être prévisible vu que le signal fourni par le capteur a_z est relativement constant (ce capteur mesure l'accélération due à la gravité) pendant tout le trajet. En outre, le signal fourni par le capteur a_z est le plus influencé par les vibrations du moteur.

Le tableau 5.5 résume les résultats des taux de bonne classification pour chaque séquence (sauf séquences 6 et 7) en utilisant SE_8 . On note une amélioration des résultats obtenus avec le sous-ensemble (SE_8) comparé à ceux obtenus avec les six capteurs. Cette amélioration peut-être vu sur le tableau 5.6. On peut voir que le taux de bonne classification global augmente de 86.5% (obtenu à l'aide de six capteurs) à 90.4% (obtenus en utilisant le sous-ensemble (SE_8)).

Table 5.5: Le taux de bonne classification obtenu en utilisant le sous-ensemble SE_8

	1	2	3	4	5	6	7	8	9	10	11
CHMMs (%)	92	100	85.7	100	89.4	-	-	83.3	88	90.9	84

Table 5.6: Les résultats obtenus en terme de F_1 -mesure par classe, F_1 -mesure global et le taux de bonne classification global (R) et son écart type (Std) pour les CHMMs en utilisant le sous-ensemble SE_8

	F_1 -mesure par classe				F_1 -mesure	TR
	LD	VG	VD	RP		$(R) \pm (Std)$
CHMMs	93.6	83.2	91.8	76.1	87.3	90.2 ± 6.1

Pour une analyse plus fine des résultats obtenus, nous nous sommes fondés sur la matrice de confusion globale. Le tableau 5.7 représente la matrice de confusion globale obtenue avec le modèle des CHMMs prenant comme variable d'intérêts le sous-ensemble de capteurs sélectionné SE_8 . Les résultats du tableau 5.7 montrent que la confusion se produit entre les paires (LD, RP), (LD, VG) et (VG, RP). La confusion entre les LD et les RP découlent du fait que ces deux situations sont similaires d'un point de vue du comportement dynamique de conduite de la moto (dans certains RP le conducteur traverse directement le RP sans suivre le marquage sur le sol). Concernant les confusions entre les paires (VG, VD) et (VG, RP), elles sont dues au fait que la durée des situations de VG sont très courtes (environ une seconde dans la plupart des situations rencontrées) comparée à la durée des autres situations, ce qui affecte la

représentativité de cette classe dans notre base de données. Ceci peut également affecter les performances du classifieur car il ne peut pas capturer les particularités de ce modèle de conduite. Ajouter à cela, les erreurs de labellisation qui peuvent être commises dans la phase d'annotation. Toutefois, les résultats dressés dans le tableau 5.7 montrent que les CHMMs arrivent à bien reconnaître les LD, VD, RP avec respectivement 91,2 %, 91 % et 100 %.

Table 5.7: Matrice de confusion globale en utilisant les six capteurs.

		classes obtenues			
		LD	VG	VD	RP
Vraies classes	LD (%)	83.8	4.4	4.2	7.6
	VG (%)	0	75	10.7	14.2
	VD (%)	1.6	3.3	91.5	3.6
	RP (%)	0	0	0	100

Table 5.8: Matrice de confusion globale en utilisant le sous-ensemble SE_8 .

		classes obtenues			
		LD	VG	VD	RP
Vraies classes	LD (%)	91.2	1.7	3.5	3.5
	VG (%)	7.1	78.5	0	14.2
	VD (%)	3.5	1.7	91	3.5
	RP (%)	0	0	0	100

En comparant les deux matrices de confusion avec et sans sélection de capteurs (tableau 5.7 et le tableau 5.8), on constate une amélioration du taux de bonne classification de chaque situation de conduite quand le sous-ensemble SE_8 est utilisé comme entrée du classifieur. En outre, la confusion entre VD et VG est presque négligeable (1,8 % de VD sont considérés comme des VG et 0 % de VG sont considérés comme VD) lors de l'utilisation du sous-ensemble SE_8 . Cependant, les erreurs sont plus importantes (3,4 % des VD sont considérés comme des VG, et 10,7 % de VG sont considérés comme de VG) quand l'ensemble des six capteurs est utilisé comme variable d'intérêt. Cette confusion s'explique par le bruit engendré par l'accélération verticale a_z .

Évaluation de la variabilité inter-sujets

Nous avons également évalué la variabilité inter-sujets. A cet effet, pour le test nous avons utilisé les données issues des séquences réalisées par un conducteur. En d'autres termes, l'apprentissage des modèles est effectué sur la concaténation des séquences réalisées par trois conducteurs et le test est effectué sur les séquences réalisées par le conducteur restant et cette procédure est répétée quatre fois. Comme on peut le voir

dans le tableau 5.9, les résultats obtenus sont relativement proches de ceux obtenus lors de l'utilisation du leave one out séquence (les taux de bonne classification globaux et leur écart type dans les deux cas est de 89.3 ± 6.1 dans le cas d'étude de la variabilité inter-sujets et 90.2 ± 5.5 dans le cas du leave one out séquence). Ce résultat signifie que la variabilité inter-sujets est négligeable dans ce cas d'étude. En d'autres termes, les situations de conduite que nous cherchons à reconnaître sont presque les mêmes d'un sujet à l'autre et ne sont pas affectées par les conditions de l'expérience (la circulation routière et les conditions météorologiques).

Table 5.9: Le taux de bonne classification obtenu par les CHMMs en utilisant le sous-ensemble SE_8 (leave one out subject)

	1	2	3	4	5	6	7	8	9	10	11
CHMMs (%)	90	98	86	97	89.1	-	-	81	88	90.9	84

5.8 Conclusion

Dans ce chapitre, nous avons présenté les approches de classification supervisée (les modèles de mélange de gaussiennes, les modèles des k plus proches voisins, les supports à vaste marge, les forêts aléatoires et les modèles de Markov cachés dans les deux cas continus est discrets) en les remettant dans le contexte applicatif de la reconnaissance d'activités de conduite des deux roues motorisés. Nous avons ainsi comparé ces différentes approches de classification pour la reconnaissance de quatre situations de conduite: ligne droite, virage à gauche, virage à droite, rond point à partir des données issues d'une centrale inertielle (accéléromètre/gyromètre).

Les résultats obtenus montrent l'efficacité de l'approche des HMMs dans les deux cas continu et discret. Le principal avantage de ce type de modèle est la prise compte de l'aspect séquentiel et de l'évolution temporelle des données, ce qui est le cas des situations de conduite. Par conséquent, ce modèle est mieux adapté pour le traitement de données temporelles du fait qu'il inclue le temps comme une variable intrinsèque. Nous avons proposé pour ce type d'application une procédure de sélection de capteur à l'aide CHMMs pour sélectionner les capteurs les plus discriminants. Cinq capteurs (a_x , a_y , r_x , r_y , r_z) sont sélectionnés comme les capteurs les plus pertinents. L'utilisation de ces capteurs permet d'améliorer le taux de bonne classification.

Chapitre 6

Détection des évènements critiques dans la conduite d'un 2RM

Sommaire

6.1	Introduction	80
6.2	Détection et classification hors ligne des évènements critiques dans la conduite d'un 2RM	80
6.2.1	Formalisation du problème de détection et de classification hors ligne des évènements critiques de conduite	81
6.2.2	Segmentation de signaux multidimensionnels via le modèle RHLP	82
6.2.3	Stratégie de détection et de classification des évènements critiques de conduite	86
6.3	Détection en ligne des chutes des 2RMs	87
6.3.1	État de l'art sur les systèmes de détection de chute des 2RMs	88
6.3.2	État de l'art sur la détection d'anomalies	89
6.3.3	Formalisation de la détection de chute à l'aide des cartes de contrôle	90
6.3.4	Description de la carte de contrôle multivariée à somme cumulée	92
6.4	Application de la méthodologie proposée et résultats	93
6.4.1	Résultats de la détection et la classification des évènements critiques de conduite	93
6.4.2	Résultats de la détection de chute via la carte de contrôle MCUSUM	99
6.5	Conclusion	104

6.1 Introduction

L'objectif de ce chapitre est de présenter une méthodologie originale permettant de détecter les événements critiques dans la conduite d'un 2RM à partir des données collectées via des capteurs embarqués (accéléromètres/gyromètres). La première partie est consacrée au problème de détection hors ligne d'événements critiques dans l'épisode de conduite d'un 2RM et la classification de séquences de conduite. Quant à la seconde partie, elle traite le développement d'un nouvel outil de détection de chute des 2RMs en utilisant les cartes de contrôle dans la perspective d'une implémentation en ligne.

6.2 Détection et classification hors ligne des événements critiques dans la conduite d'un 2RM

Les erreurs de conduite sont les principales causes des accidents des véhicules en général et en particulier des 2RMs. Cette particularité réside du fait que la conduite d'un 2RM demande plus d'implication du conducteur dans le contrôle de son véhicule. Par conséquent, observer, comprendre et analyser le comportement du conducteur pendant ou juste avant l'occurrence de ces événements critiques¹ dans l'épisode de conduite (accident et presque accident) a fait l'objet de plusieurs travaux de recherche. De nombreuses études ont été menées pour déterminer les facteurs associés à ces événements critiques qui se produisent généralement avant les accidents. Nous pouvons citer les études sur l'analyse qualitative des données d'accidentologie comme les rapports de la gendarmerie, ou encore les enquêtes basées sur le déclaratif des accidentés etc. Toutefois, ces données sont collectées sur la base des témoignages des accidentés. Elles sont potentiellement partielles et entachées d'incertitudes, vu qu'elles sont recueillies sous l'effet de l'émotion après l'accident. L'inconvénient majeur de ce type d'approche réside dans l'usage de données subjectives pouvant affecter cette analyse.

Dans le but de s'affranchir de cette subjectivité, plusieurs projets de recherche ont été récemment mis en place. Ces projets ont pour objectif d'étudier les comportements des conducteurs de 2RMs dans un contexte de conduite réelle dite naturelle. Des 2RMs équipés de capteurs embarqués ont été utilisés afin de collecter une grande quantité de données liées à la conduite naturelle des 2RMs. Dans ce contexte, l'analyse visuelle des données issues de ces expérimentations peut s'avérer fastidieuse, surtout pour l'identification des événements critiques de conduite. D'où la nécessité de développer des outils d'analyse automatique à cet effet.

Dans la littérature, il y a peu de travaux qui ont abordé le problème de détection automatique des événements critiques dans la conduite d'un 2RM. La seule étude qui

¹Un événement critique est une situation de conduite durant laquelle un accident a peut-être été évité.

porte sur ce problème a été réalisée dans le cadre du projet 2BESAFE [69]. La détection des évènements critiques de conduite s'est alors basée sur une métrique (distance de Mahalanobis) qui permet de détecter un décentrage par rapport à la moyenne. Les auteurs ont appliqué leur approche sur une base de données issues d'une expérimentation de conduite réelle. Trois ensembles de signaux issus de capteurs ont été évalués :

- Le premier ensemble est constitué de signaux issus des capteurs de contrôle commande (angle guidon, position de l'accélérateur et l'état du frein);
- Le deuxième ensemble est constitué de signaux issus des capteurs sur la dynamique de la moto (accélération et vitesse linéaire)
- Le troisième ensemble est constitué de signaux issus de tous les capteurs (premier et deuxième ensemble).

Cette méthode présente quelques inconvénients, à savoir:

- la détection des évènements critiques de conduite est obtenue en utilisant les capteurs de contrôle commande. L'instrumentation des motos avec ce type de capteur peut entraîner des modifications des propriétés dynamiques de la moto (problème d'homologation).
- La validation de cette méthode a été réalisée sur des données collectées avec un seul conducteur; ce qui pose des problèmes de généralisation de la méthode due à des problèmes de variabilités des observations. Cette capacité de généralisation est fortement dépendante du choix du seuil de détection fixé de manière empirique.
- L'utilisation de la distance de Mahalanobis ne permet pas de tenir compte des états antérieurs du système de conduite.

6.2.1 Formalisation du problème de détection et de classification hors ligne des évènements critiques de conduite

L'objectif de cette partie du chapitre est de présenter une méthodologie de détection et de classification hors ligne des évènements critiques dans la conduite des 2RMs à partir de données collectées via des capteurs embarqués (accéléromètres/gyromètres). Le choix d'une telle instrumentation est lié au fait que ces capteurs ne provoquent pas de modification sur la dynamique de la moto d'une part et, d'autre part, ce type de capteurs permet de mesurer toutes les variations dynamiques de la moto.

Nous avons formulé le problème de détection des évènements critiques de conduite comme un problème de détection de changement des statistiques (moyennes et variances) des signaux issus des capteurs d'un régime de "conduite normale" à un régime

de "conduite critique". Pour cela, on cherche à segmenter les signaux issus des capteurs d'une séquence de conduite et estimer ensuite les moyennes et les variances de ces derniers pour chaque segment. La variation des moyennes et des variances de deux régimes successifs est exploitée pour quantifier l'importance du changement. La détection et la classification des évènements critiques de conduite est ainsi réalisée en deux étapes:

- La première étape consiste en la segmentation non supervisée des séries temporelles multidimensionnelles (accéléromètres/gyromètres) d'une séquence de conduite avec estimation de la moyenne et de la variance de chaque segment. En d'autres termes, chaque segment est modélisé par le couple $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, avec $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ représentent respectivement le vecteur des moyennes et la matrice de variance-covariance des accéléromètres/gyromètres.

A cet effet, un mélange de densités gaussiennes à proportions logistiques est proposé dans ce chapitre. Ce modèle est un cas particulier du modèle de régression logistique à processus cachés (de l'anglais Regression model with a Hidden Logistic Process [70]) qui sera décrit dans la suite du chapitre. Ce modèle sera comparé à un modèle de Markov caché (HMM).

- La deuxième étape consiste à affecter chaque segment à la classe de conduite "normale" ou bien à la classe de conduite "critique" (chute ou presque chute) en exploitant les paramètres des segments. A cet effet, l'algorithme des k -plus proches voisins est utilisé.

6.2.2 Segmentation de signaux multidimensionnels via le modèle RHLP

Dans ce paragraphe, nous nous intéresserons à la description du modèle RHLP (Regression model with a Hidden Logistic Process) [70], dans ses versions univariée et multivariée.

Rappels sur le modèle RHLP: cas univarié

Le modèle RHLP est un mélange de régression polynomiale dans lequel, la transition (passage) d'un modèle de régression à un autre est contrôlée par un processus logistique caché. Le modèle RHLP utilise un processus logistique pour modéliser une séquence d'observations cachée. Pour la version univariée, le modèle RHLP de la séquence des variables observées (x_1, x_2, \dots, x_N) est défini par:

$$x_j = \boldsymbol{\beta}_{z_j}^T \mathbf{t}_j + \sigma_{z_j} \epsilon_j; \epsilon_j \sim \mathcal{N}(0, 1), j = 1, \dots, N, \quad (6.1)$$

où la variable z_j commande le passage d'un modèle de régression à l'autre, à chaque instant t_j ; on a $\beta_{z_j} = (\beta_{z_j}^0, \dots, \beta_{z_j}^p)^T$, le vecteur des coefficients de régression polynomiale du modèle ; $\mathbf{t}_j = (1, t_j^1, t_j^2, \dots, t_j^p)^T$ est le vecteur de covariable de dimension $p + 1$ à l'instant t_j avec p qui représente l'ordre de régression. Le modèle RHLP suppose que la séquence cachée $\mathbf{z} = (z_1, \dots, z_N)$ est un processus logistique caché, avec $z_j \in \{1, \dots, K\}$. On suppose que les variables z_j étant donné le vecteur $\mathbf{t} = (t_1, \dots, t_N)$ sont générées de façon indépendante selon la distribution multinomiale $\mathcal{M}(1, \pi_1(t_j, \mathbf{w}), \dots, \pi_K(t_j, \mathbf{w}))$, où

$$\pi_k(t_j; \mathbf{w}) = p(z_j = k | t_j; \mathbf{w}) = \frac{\exp(\mathbf{w}_k^T \mathbf{v}_j)}{\sum_{l=1}^K \exp(\mathbf{w}_l^T \mathbf{v}_j)}, \quad (6.2)$$

est la transformation logistique d'une fonction linéaire du vecteur des covariables de temps $\mathbf{v}_j = (1, t_j, t_j^2, \dots, t_j^u)^T$; $\mathbf{w}_k = (\mathbf{w}_{k0}, \dots, \mathbf{w}_{ku})^T$ est le vecteur des coefficients de dimension $(u + 1)$ associés à \mathbf{v}_j et $\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_K)$. Ainsi, étant donné le vecteur $\mathbf{t} = (t_1, \dots, t_N)$, la distribution de \mathbf{z} peut s'écrire sous la forme:

$$p(\mathbf{z} | \mathbf{t}; \mathbf{w}) = \prod_{j=1}^N \prod_{k=1}^K p(z_j = k | t_j; \mathbf{w})^{z_{jk}} = \prod_{j=1}^N \prod_{k=1}^K \left[\frac{\exp(\mathbf{w}_k^T \mathbf{v}_j)}{\sum_{l=1}^K \exp(\mathbf{w}_l^T \mathbf{v}_j)} \right]^{z_{jk}}, \quad (6.3)$$

où z_{jk} est une variable binaire qui vaut 1 lorsque \mathbf{x}_j est générée par le $k^{ième}$ modèle de régression, et qui vaut 0 sinon.

Extension multidimensionnelle du modèle RHLP dédié à la détection d'évènements critiques

Dans le cadre de notre travail pour la détection et la classification des événements critiques de conduite, nous nous sommes focalisés sur une version multivariée du modèle RHLP à deux composantes gaussiennes ($K=2$), non régressives ($p=0$). Le choix d'un tel modèle est justifié par le fait que la détection des événements critiques de conduite se base sur les statistiques (moyennes et variances) des données de mesures de deux régimes de conduite successifs (conduite normale et conduite critique). En d'autres termes, on cherche à modéliser chaque régime de conduite par une gaussienne (moyennes et variances).

Soit une série temporelle \mathbf{X} à N observations avec $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_j, \dots, \mathbf{x}_N)$ à l'instant $\mathbf{t} = (t_1, \dots, t_N)$ avec $\mathbf{x}_j = (x_j^{(1)}, x_j^{(2)}, \dots, x_j^{(d)}) \in \mathbb{R}^d$ est la $j^{ième}$ observation.

Le modèle RHLP multivariée noté MRHLP est reformulé comme un ensemble de plusieurs modèles RHLP non régressif. Dans ce cas l'observation \mathbf{x}_j à l'instant t_j est

supposée être générée par le modèle suivant:

$$\begin{aligned} x_j^1 &= \beta_{z_j}^{(1)} + \sigma_{z_j}^{(1)} \epsilon_j \\ x_j^2 &= \beta_{z_j}^{(2)} + \sigma_{z_j}^{(2)} \epsilon_j \\ &\vdots \\ x_j^d &= \beta_{z_j}^{(d)} + \sigma_{z_j}^{(d)} \epsilon_j \end{aligned} \quad (6.4)$$

où la séquence cachée $\mathbf{z} = (z_1, \dots, z_N)$ est commune à toutes les composantes univariées de la séquence d'observations. Le modèle non régressif décrit ci-dessus peut être réécrit de la manière suivante:

$$\mathbf{x}_j = \beta_{z_j} + \Sigma_{z_j} \epsilon_j; \epsilon_j \sim \mathcal{N}(0, 1), (j = 1, \dots, N), \quad (6.5)$$

où $\mathbf{x}_j = (x_j^{(1)}, \dots, x_j^{(d)})^T \in \mathbb{R}^d$ est la $j^{\text{ième}}$ observation, $\beta_{z_j} = [\beta_{z_j}^1, \dots, \beta_{z_j}^d]$ est le vecteur des paramètres du modèle associés au régime (segment) $z_j = k$. Il est de dimension $1 \times d$ et Σ_{z_j} est la matrice diagonale de variance correspondante.

Ce modèle peut être représenté graphiquement comme sur la figure 6.1.

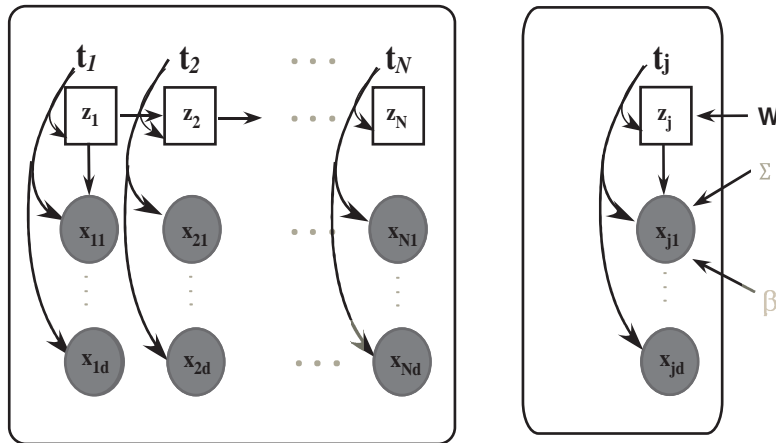


Figure 6.1: Modèle graphique de génération de données d'un modèle MRHLP.

Le modèle de segmentation utilisé dans ce travail suppose que chaque vecteurs d'observations \mathbf{x}_t est distribué selon le mélange de deux distributions gaussiennes défini par:

$$p(\mathbf{x}_t; \boldsymbol{\theta}) = \pi(\mathbf{t}; \mathbf{w}) \mathcal{N}(\mathbf{x}_t; \beta_1, \Sigma_1) + (1 - \pi(\mathbf{t}; \mathbf{w})) \mathcal{N}(\mathbf{x}_t; \beta_2, \Sigma_2) \quad (6.6)$$

où $\mathcal{N}(\cdot; \beta, \Sigma)$ est la fonction de densité de probabilité avec la moyenne β et la matrice de covariance Σ . Les paramètres $(\beta_{z_t}, \Sigma_{z_t})_{z_t=1,2}$ sont les moyennes et les matrices de covariance des composantes du mélange et $\pi(\mathbf{t}; \mathbf{w})$ est la première proportion du

mélange définie par :

$$\pi(\mathbf{t}; \mathbf{w}) = \frac{\exp(w_0 + w_1 t + w_2 t^2)}{1 + \exp(w_0 + w_1 t + w_2 t^2)}. \quad (6.7)$$

où le paramètre $\mathbf{w} = (w_0, w_1, w_2) \in \mathbb{R}^3$. Le paramètre $\boldsymbol{\theta} = \{\mathbf{w}, \beta_1, \beta_2, \Sigma_1, \Sigma_2\}$ de ce modèle est estimé en maximisant la log-vraisemblance via l'algorithme EM:

La log-vraisemblance du modèle de données observées \mathbf{x}_j est donnée par:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{t}) &= \log \prod_{j=1}^N p(\mathbf{x}_j | t_j; \boldsymbol{\theta}) \\ &= \sum_{j=1}^N \sum_{k=1}^K \pi_k(t_j, \mathbf{w}) \mathcal{N}(\mathbf{x}_j, \beta_k, \Sigma_k) \end{aligned} \quad (6.8)$$

La maximisation de cette log-vraisemblance ne peut être effectuée analytiquement car il en résulte une fonction non-linéaire en raison du logarithme de la somme. Cependant, dans le contexte des variables latentes, la maximisation par l'algorithme EM est particulièrement adaptée. Dans ce cas, la log-vraisemblance associée aux données complétées est donnée par :

$$\mathcal{L}_c(\boldsymbol{\theta}; \mathbf{X}, \mathbf{z}, \mathbf{t}) = \log(p(\mathbf{x}, \mathbf{z} | \mathbf{t}; \boldsymbol{\theta})) = \log[p(\mathbf{x} | \mathbf{z}, \mathbf{t}; \boldsymbol{\theta}) p(\mathbf{z} | \mathbf{t}; \mathbf{w})]. \quad (6.9)$$

Ainsi, cette log-vraisemblance peut s'écrire de la façon suivante:

$$\mathcal{L}_c(\boldsymbol{\theta}; \mathbf{X}, \mathbf{z}, \mathbf{t}) = \sum_{j=1}^N \sum_{k=1}^K z_{jk} \log[\pi_k(t_j, \mathbf{w}) \mathcal{N}(\mathbf{x}_j, \beta_k, \Sigma_k)]. \quad (6.10)$$

L'algorithme EM commence avec un paramètre initial $\boldsymbol{\theta}^{(0)}$ et répète les étapes E et M suivantes jusqu'à la convergence:

l'étape E: Elle consiste à calculer l'espérance de la log-vraisemblance des données complétées:

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}) &= \mathbb{E}[\mathcal{L}_c(\boldsymbol{\theta}; \mathbf{X}, \mathbf{z}, \mathbf{t}) | \mathbf{X}, \mathbf{t}; \boldsymbol{\theta}^{(q)}] \\ &= \sum_{j=1}^N \sum_{k=1}^K \tau_{jk}^{(q)} \log \pi_k(t_j, \mathbf{w}) + \sum_{j=1}^N \sum_{k=1}^K \tau_{jk}^{(q)} \log \mathcal{N}(\mathbf{x}_j, \beta_k, \Sigma_k) \end{aligned} \quad (6.11)$$

Comme le montre l'expression de la fonction Q , cette étape nécessite simplement le calcul des probabilités *a posteriori* $\tau_{jk}^{(q)}$:

$$\begin{aligned}\tau_{jk}^{(q)} &= p(z_{jk} = 1 | \mathbf{x}_j, t_j; \boldsymbol{\theta}^{(q)}) \\ &= \frac{\pi_k(t_j, \mathbf{w}^{(q)}) \mathcal{N}(\mathbf{x}_j, \boldsymbol{\beta}_k^{(q)}, \boldsymbol{\Sigma}_k^{(q)})}{\sum_{l=1}^K \pi_l(t_j, \mathbf{w}^{(q)}) \mathcal{N}(\mathbf{x}_j, \boldsymbol{\beta}_l^{(q)}, \boldsymbol{\Sigma}_l^{(q)})}\end{aligned}\quad (6.12)$$

l'étape M: Elle permet de mettre à jour les paramètres du modèle par le calcul de $\boldsymbol{\theta}^{(q+1)}$ en maximisant la fonction Q par rapport à $\boldsymbol{\theta}$. On obtient les formules de mise à jour suivantes [70] :

La formule de mise à jour de $\boldsymbol{\beta}_k$ est donnée par:

$$\boldsymbol{\beta}_k^{(q+1)} = \frac{\sum_{j=1}^N \tau_{jk}^{(q)} \mathbf{x}_j}{\sum_{j=1}^N \tau_{jk}^{(q)}} \quad (6.13)$$

La formule de mise à jour de $\boldsymbol{\Sigma}_k$ est donnée par:

$$\boldsymbol{\Sigma}_k^{(q+1)} = \frac{\sum_{j=1}^N \tau_{jk}^{(q)} [\mathbf{x}_j - \boldsymbol{\beta}_k^{(q+1)}][\mathbf{x}_j - \boldsymbol{\beta}_k^{(q+1)}]^T}{\sum_{j=1}^N \tau_{jk}^{(q)}} \quad (6.14)$$

La maximisation de $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$ par rapport à \mathbf{w} est un problème de régression logistique multinomiale pondérée par $\tau_{jk}^{(q)}$ qui se résout via l'algorithme IRLS (de l'anglais Iteratively Reweighted Least Squares).

La formule de mise à jour de \mathbf{w} est donnée par:

$$\mathbf{w}_k^{(q+1)} = \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{j=1}^N \sum_{k=1}^K \tau_{jk}^{(q)} \log[\pi_K(t_j, \mathbf{w}^{(q)})] \quad (6.15)$$

Le pseudo code 5 résume les différentes étapes de l'algorithme EM dédié à l'estimation des paramètres du modèle MRHLP.

6.2.3 Stratégie de détection et de classification des évènements critiques de conduite

Comme décrit précédemment, notre stratégie de détection et de classification des évènements de conduite est réalisée en deux étapes (la segmentation et la classification).

- **Segmentation des séquences de conduite**

Pour la segmentation des séquences de conduite, nous avons formulé le problème comme une régression multidimensionnelle d'ordre 0 à partir des données accéléromètres/gyromètres $\mathbf{x}_t^l = \{a_x, a_y, a_z, r_x, r_y, r_z\}$ de chaque séquence de conduite l , avec $l = [1, 2, \dots, 29]$ et $t \in [1, \dots, T^{(l)}]$.

Nous avons mis en concurrence le modèle proposé avec un modèle de Markov

Algorithme 5

Entrées: Une séquence d'observations $\mathbf{X} = (\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_N)$

Le nombre de polynômes K

Le degré de régression p

Fixer un seuil $\epsilon > 0$

Mettre $q = 0$

Sorties: $\hat{\boldsymbol{\theta}}$ // ensemble des paramètres du modèle

```

1: Initialisation:
2:  $\hat{\boldsymbol{\theta}}^{(0)}$ ;
3: tantque incrément de Log-vraisemblance  $> \epsilon$  faire
4:   Étape E
5:   Calculer les quantités  $\tau_{jk}^{(q)}$  en utilisant l'équation (6.12)
6:   Étape M
7:   Calculer  $\mathbf{w}^{(q+1)}$  en utilisant l'équation (6.15)
8:   pour  $k=1, \dots, K$  faire
9:     Calculer  $\beta_k^{(q+1)}$  en utilisant l'équation (6.13)
10:    Calculer  $\Sigma_k^{(q+1)}$  en utilisant l'équation (6.14)
11:   fin pour
12:    $q \leftarrow q + 1$ 
13: fin tantque
14:  $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(q)}$ 

```

Outputs: $\hat{\boldsymbol{\theta}}$.

caché continu (HMM). Pour cela, nous avons utilisé un modèle HMM gauche-droite à deux états cachés. Dans ce modèle, chaque état émet une observation \mathbf{x}_t^l selon un mélange de deux gaussiennes comme dans le cas précédent.

- **Classification des segments extraits des signaux**

Après cette étape de segmentation, un ensemble de caractéristiques est calculé en fonction du paramètre $\boldsymbol{\theta}$ estimé pour chaque séquence. L'ensemble des caractéristiques \mathcal{F} est calculé comme suit:

$$\mathcal{F} = \{\beta_1, \beta_2, \|\beta_1 - \beta_2\|, \text{diag}(\boldsymbol{\Sigma}_1), \text{diag}(\boldsymbol{\Sigma}_2), \|\text{diag}(\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_2)\|\}$$

où $\|\cdot\|$ est la norme associée à la distance Euclidienne entre β_1 et β_2 , et $\text{diag}(\boldsymbol{\Sigma})$ est le vecteur des éléments diagonaux de $\boldsymbol{\Sigma}$. Cet ensemble de caractéristiques est utilisé comme base d'apprentissage pour classifier les segments en utilisant l'algorithme des k -plus proches voisins.

6.3 Détection en ligne des chutes des 2RMs

De nos jours, les systèmes de protection installés sur les 2RMs sont pour la plupart des systèmes de sécurité active. Nous avons le système anti-blocage des roues, plus

connu sous l'abréviation ABS (de l'allemand Anti Blockier System) où le système de freinage Combiné plus connu sous l'abréviation CBS (de l'anglais Combined Brake System). Ces systèmes ne sont disponibles que sur une certaine gamme de motos (haut de gamme), vu leur coût élevé d'installation de ce type de système. Par ailleurs, peu de systèmes de sécurité passive ont été développés pour les 2RMs, mis à part les équipements du conducteur tels que les vêtements et en particulier les casques et plus récemment l'airbag. Ce dernier a été mis au point par la société Honda [71, 72, 73] et n'est proposé actuellement que sur un véhicule «haut de gamme» de la marque.

Récemment, le gilet airbag a été développé et mis sur le marché depuis moins de dix ans dans sa version filaire. Le conducteur via le gilet est relié à la moto par un câble. En cas de chute, dès que le conducteur est éjecté de son véhicule, la tension exercée sur le câble actionne un mécanisme qui déclenche le gonflement du coussin de sécurité. Une étude réalisée à l'IFSTTAR a montré qu'un airbag filaire, de par sa conception, se montre peu efficace dans certains types de chocs du fait du délai de mise en action trop long. D'où l'intérêt des gilets airbag à système électronique (activation du gonflage du gilet par un signal radio) qui consiste à détecter des chutes où des chocs en se fondant sur des capteurs accélérométriques installés sur la moto. La réaction de ce type de système se décompose en trois phases : (1) la détection de l'accident; (2) l'activation du système du gonflage; (3) le gonflage du coussin de protection. Une activation trop précoce ou trop tardive peut être une source de risque (la durée typique de gonflage du coussin de protection étant de 100 ms [74]. Par conséquent, toute la problématique de ce type de système de sécurité réside dans le moment de la détection des chutes ou des chocs.

6.3.1 État de l'art sur les systèmes de détection de chute des 2RMs

De nombreux travaux de recherche ont été menés pour développer un système de détection de chutes dédié aux 2RMs en utilisant des capteurs embarqués. Certaines de ces études ont été réalisées avec des équipes de course [75, 76, 77]. Cependant, la fonction de détection de chute utilisée dans ces études reste confidentielle. Dans [78], les auteurs ont proposé un algorithme de détection des événements dangereux (dits accidentogènes) en évaluant le degré de corrélation entre l'angle de lacet nominal reconstruit à partir de l'angle du guidon et du modèle de véhicule, et l'angle de lacet mesuré par le gyromètre. Lorsque le degré de corrélation dépasse un certain seuil, la situation est dite accidentogène. Dans [74] un algorithme de détection des chutes basé sur le calcul de la norme de l'accélération et de la vitesse angulaire sur les trois axes (X-Y-Z) a été développé. Cette méthode est basée sur le calcul de la norme du vecteur accélérations $\|\mathbf{A}\|$ ainsi que la norme du vecteur vitesses angulaires $\|\mathbf{R}\|$ dans l'espace à trois dimensions. Ces

normes sont calculées en utilisant les équations suivantes:

$$\|\mathbf{A}\| = \sqrt{\mathbf{a}_x^2 + \mathbf{a}_y^2 + \mathbf{a}_z^2}. \quad (6.16)$$

$$\|\mathbf{R}\| = \sqrt{\mathbf{r}_x^2 + \mathbf{r}_y^2 + \mathbf{r}_z^2}. \quad (6.17)$$

A chaque instant, on calcule les deux normes $\|\mathbf{A}\|$ et $\|\mathbf{R}\|$ et on les compare à des seuils choisis empiriquement. Ainsi, la chute est détectée lorsque ces deux seuils sont franchis. Pour plus de détails, voir [74].

L'objectif de cette partie du chapitre est de présenter un nouvel outil de détection en ligne de chute des 2RMs à partir des données collectées via des capteurs embarqués (accéléromètres/gyromètres). Nous avons formulé ce problème comme un problème de détection séquentielle d'anomalie.

6.3.2 État de l'art sur la détection d'anomalies

La détection d'anomalies se réfère au problème de recherche de motifs dans des données qui ne sont pas conformes au comportement "attendu". Selon le domaine d'application, ces données non conforme sont souvent désignées comme des anomalies, valeurs aberrantes, observations discordantes. L'utilisation de la détection d'anomalies est une problématique traitée dans une grande variété d'applications telles que la détection de la fraude des cartes de crédit, d'assurance ou de soins de santé [79, 80, 81], la détection d'intrusion pour la cybercriminalité [82], la détection des défauts dans les systèmes critiques pour la sécurité et la surveillance [83].

Dans la littérature, la détection séquentielle d'anomalies peut être fondée sur des techniques de reconnaissance de formes. La classification mono-classe est généralement utilisée [84, 85, 86, 87, 88]. D'autres méthodes fondées sur la notion du voisinage ont été aussi proposées [88, 89, 90]. On peut citer également les travaux de [91, 92] sur la détection des points de changement par l'utilisation des modèles de Markov cachés. Un panorama de méthodes utilisées dans ce domaine peut être trouvé dans [93]. Dans la détection séquentielle d'anomalies fondée sur les tests d'hypothèses, on suppose que dans une séquence d'observations, une partie des observations suit une distribution P_{θ_0} et une autre partie suit une distribution P_{θ_1} . On appelle t le temps de changement, en d'autres termes, le temps à partir duquel la distribution des observations change de P_{θ_0} à P_{θ_1} .

Le problème de détection d'anomalies peut être ainsi définie: Soit $\mathbf{x} = (x_1, x_2, \dots, x_t)$, une séquence complète d'observations. Supposons que la séquence d'observations $(x_1, x_2, \dots, x_{t_c-1})$ suit une distribution P_{θ_0} et la séquence d'observations (x_{t_c}, \dots, x_t) suit une distribution P_{θ_1} , où t_c représente le temps de changement de la distribution. Il est souvent supposé que le processus de données avant et après le changement est sta-

tionnaire. L'objectif est d'établir un test binaire qui permet de détecter le changement de régime. Les hypothèses du test s'expriment de la manière suivante:

$$\begin{cases} H_0 : x_i \sim P_{\theta_0} & \forall i = 1, \dots, t \\ H_1 : \begin{cases} : x_i \sim P_{\theta_0} & \forall i = 1, \dots, x_{t_c-1} \\ : x_i \sim P_{\theta_1} & \forall i = x_{t_c}, \dots, x_t \end{cases} \end{cases} \quad (6.18)$$

où H_0 représente l'hypothèse nulle (pas de changement) et H_1 représente l'hypothèse alternative (avec changement).

D'après [94] on distingue deux cadres de détection d'anomalies:

- **Détection hors ligne** où il s'agit de détecter le point de changement en se basant sur toute la séquence d'observation x .
- **Détection en ligne** où l'objectif est de détecter le point de changement au fur et à mesure que les observations arrivent en se basant sur les données disponibles. C'est cette dernière que nous retiendrons dans notre cas d'étude.

Quand l'hypothèse de changement H_1 est acceptée, la règle de décision est souvent donnée par:

$$t_\alpha = \inf\{t \geq 1 : g_t \geq h\}, \quad (6.19)$$

où t_α représente le temps de déclenchement de l'alarme, g_t la statistique du test à l'instant t et h le seuil de détection fixé par l'utilisateur. Le choix de h dépend essentiellement du taux de fausses alarmes.

6.3.3 Formalisation de la détection de chute à l'aide des cartes de contrôle

La maîtrise statistique des processus (MSP) en anglais (Statistical Process Control (SPC)) englobe des méthodes utilisées pour surveiller et améliorer les procédés de fabrication industrielle. L'un des outils les plus utilisés en MSP est les cartes de contrôle. Ces outils ont montré leur efficacité dans plusieurs domaines, notamment pour le système de freinage des bus [95], pour l'aviation [96], pour l'ingénierie [97, 98], [99] Les cartes de contrôle généralement utilisées sont les cartes de Shewhart (Walter A. Shewhart) qui se basent sur la détection de variations anormales du décentrage ou de la dispersion des données supposées suivre une distribution normale $\mathcal{N}(\mu_0, \sigma_0)$ dite sous contrôle. Un processus est dit sous contrôle, si ce dernier évolue dans les limites de contrôle supérieur et inférieur, respectivement UCL (Upper Control Limit) et LCL (Lower Control Limit). Les limites de contrôle sont généralement calculées comme suit:

$$\begin{cases} UCL = \mu_0 + \lambda\sigma_0 \\ LCL = \mu_0 - \lambda\sigma_0 \end{cases} \quad (6.20)$$

avec σ_0 l'écart type des données sous contrôle, μ_0 la moyenne des données sous contrôle et λ un facteur multiplicatif.

Dans la mise en application des cartes de contrôle, il est utile de distinguer les deux phases d'utilisation [94]:

- **La phase I** dite phase d'observation, on analyse un ensemble de données de distribution sous contrôle inconnu. Les objectifs de cette phase sont de comprendre la variation d'un processus au fil du temps, d'évaluer sa stabilité et de modéliser les performances dans la maîtrise du processus. Cette dernière étape est en général réalisée par l'estimation des paramètres d'un modèle paramétrique.
- **La phase II** dite phase de surveillance, on s'intéresse à la surveillance des processus en utilisant des données collectées séquentiellement, afin de détecter le plus tôt possible les changements dans le processus de surveillance. Dans cette phase la distribution sous contrôle peut être connue ou inconnue.

Dans la mise en œuvre d'une carte de contrôle, il est important d'estimer le paramètre $\theta_0 = (\mu_0, \sigma_0)$, lorsque celui-ci est inconnu.

Pour ce faire, soit une séquence d'observations (x_1, x_2, \dots, x_t) qui est sous-échantillonnée en sous-groupes de taille W comme suit:

$$\bar{x}_i = \frac{1}{W} \sum_{j=0}^{W-1} x_{i.W-j}, \quad \text{avec } i \in \{1, 2, \dots, n_t\} \quad (6.21)$$

Ensuite, on calcule la matrice de variance correspondante:

$$\bar{S}_i = \frac{1}{W-1} \sum_{j=0}^{W-1} (x_{i.W-j} - x_i)(x_{i.W-j} - x_i)^T, \quad \text{avec } i \in \{1, 2, \dots, n_t\}. \quad (6.22)$$

où i représente le $i^{\text{ième}}$ sous-groupe de taille W . On obtient ainsi n_t sous-groupes $(W, 2W, \dots, n_t W)$, avec $n_t = \frac{t}{W}$.

Ainsi pour un ensemble d'échantillons sous contrôle, de taille t_0 , les paramètres $\hat{\theta}_0 = (\hat{\mu}_0; \hat{\sigma}_0)$ peuvent être estimés à l'aide des relations suivantes:

$$\hat{\mu}_0 = \frac{1}{n_{t_0}} \sum_{i=1}^{n_{t_0}} \bar{x}_i \quad (6.23)$$

$$\hat{\sigma}_0 = \frac{1}{n_{t_0}} \sum_{i=1}^{n_{t_0}} \bar{S}_i \quad (6.24)$$

Dans la version initiale, les cartes de contrôles étaient uni-variées; elles ne permettaient pas de prendre en compte l'aspect multivarié des données. En 1947, Hotelling introduit l'aspect multivarié afin de tenir compte de la corrélation entre les différentes variables. Dans ce cas de figure, il n'y a pas de limite inférieure ($LCL = 0$) et la limite de contrôle est la limite supérieure UCL_α paramétrée par un taux de fausses alarmes α . Parmi les cartes multi-variées les plus utilisées, on peut citer la carte de contrôle multivariée de Shewhart, la carte de contrôle MEWMA (multivariée exponentielle-moyenne mobile pondérée) et la carte de contrôle MCUSUM (Multivariate CUMulative SUM control chart) [100].

La carte de contrôle MCUSUM attire tout notre intérêt, car cette carte cumule les informations les plus récentes et les informations précédentes (la carte de contrôle MCUSUM tient compte de l'état antérieur de la procédure de détection). Elle a également la capacité de détecter la moindre variation dans les paramètres du processus. Dans la section suivante, on donnera une brève description de la carte de contrôle MCUSUM.

6.3.4 Description de la carte de contrôle multivariée à somme cumulée

La carte de contrôle de type somme cumulée (CUSUM) a été introduit par [101]. Par la suite, plusieurs travaux ont donné des versions améliorées en particulier [102, 103, 104, 105, 106, 107, 108, 109, 110]. Selon la statistique de test g_t de la carte, on peut distinguer plusieurs types de cartes CUSUM. Ici, on ne présente que la carte multivariée MCUSUM [100].

La carte multivariée CUSUM est considérée comme l'une des cartes de contrôle les plus performantes au sens de l'ARL que nous allons définir par la suite. La statistique de test g_t d'une telle carte est donnée par:

$$g_t = \sqrt{\mathbf{s}_t' \boldsymbol{\Sigma}_0^{-1} \mathbf{s}_t} \quad (6.25)$$

avec

$$\mathbf{s}_t = \begin{cases} 0 & \text{si } \mathbf{C}_t \leq k \\ (\mathbf{s}_{t-W} + \bar{x}_t - \boldsymbol{\mu}_0) \cdot (1 - \frac{k_1}{\mathbf{C}_t}) & \text{si } \mathbf{C}_t > k \end{cases}$$

Où

$$\mathbf{C}_t = \sqrt{(\mathbf{s}_{t-W} + x_t^w - \boldsymbol{\mu}_0)' \cdot \boldsymbol{\Sigma}_0^{-1} \cdot (\mathbf{s}_{t-W} + \bar{x}_t - \boldsymbol{\mu}_0)}, \quad \text{et } g_0 = 0. \quad (6.26)$$

Cette carte indique la présence d'une anomalie quand la statistique g_t dépasse un seuil h prédéfini.

6.4 Application de la méthodologie proposée et résultats

Cette partie du chapitre consiste à appliquer la méthodologie proposée sur des données de mesures issues d'une expérimentation réelle. Rappelons que la méthodologie proposée porte sur la détection et la classification des événements critiques de conduite ainsi que la détection en ligne des chutes des 2RMs. Nous avons utilisé la base de données contenant les 29 séquences de conduite décrites au chapitre 3. Pour rappel, cette base de données est constituée de 8 séquences de chute avec cinq scénarios, 10 séquences de presque chute représentant six scénarios et 11 séquences de conduite normale. Les variables d'entrées de la carte de contrôle que nous avons utilisée sont les accélérations et les vitesses de rotation $\mathbf{x}_t^l = \{a_x, a_y, a_z, r_x, r_y, r_z\}$ de chaque séquence l , avec $l \in \{1, 2, \dots, 29\}$ et $t \in [1, \dots, T^{(l)}]$.

6.4.1 Résultats de la détection et la classification des événements critiques de conduite

On présente ici les résultats de détection et de classification des événements critiques de conduite d'un 2RM, obtenus sur une base de données réelles. Pour rappel, la détermination des plages temporelles d'occurrences des événements critiques de conduite (chute et presque chute) dans une séquence est réalisée manuellement en se fondant sur les données vidéo.

Pour évaluer les performances de l'approche proposée, nous avons procédé comme suit:

- Calcul du taux de bonne segmentation des séquences de conduite (taux de bonne détection des plages temporelles d'occurrences des événements critiques de conduite pour chaque séquence de conduite) des deux approches non-supervisées (MRHLP et HMM).
- Calcul du taux de bonne classification des séquences de conduite (conduite normale, conduite avec presque chute et conduite avec chute). Pour cela, le modèle des k -ppv prenant comme variables d'entrée des caractéristiques extraites à partir des modèles de segmentation est évalué en utilisant la méthode de k -cross validation avec $k=10$.

La figure 6.2 représente un exemple de segmentation de trois séquences de conduite à savoir une séquence de conduite avec chute, une séquence de conduite avec presque chute et une séquence de conduite normale (respectivement *séquence₇*, *séquence₁₇* et

*séquence*₁₉). Les premières figures (a, b et c) représentent les données accélérométriques enregistrées durant les trois cas de figures (les accélérations a_z et a_y sont représentées avec un décalage de $+2g$ et $-2g$ respectivement). Les bandes vertes représentent les vraies plages temporelles d'occurrence des événements critiques de conduite (chute et presque chute). Les figures (d, e et f) représentent les probabilités estimées avec le modèle MRHLP respectivement pour la *séquence*₇, la *séquence*₁₇ et la *séquence*₁₉. Les dernières figures (g, h et i) représentent les probabilités estimées avec le modèle HMM pour les mêmes séquences.

Comme nous pouvons le constater la méthode de segmentation par le modèle MRHLP segmente mieux les données par rapport au modèle HMM. Dans le cas de la séquence de chute, il peut être clairement observé que les données sont correctement segmentées par le modèle MRHLP voir la figure 6.2(d). Nous pouvons également remarquer que les données sont aussi correctement segmentées par le modèle HMM avec quelques "pseudo" transitions éphémères sur la probabilité estimée (voir la figure 6.2(g)).

Dans le cas de la séquence de presque chute, le même constat peut être fait pour les deux algorithmes, avec plus de "pseudo" transitions dans le cas du modèle HMM comme illustré sur les figures 6.2 (e, h). Ces "pseudo" transitions apparaissent en particulier dans les phases de transition c.à.d, au moment de changement (de la conduite normale vers la conduite en zigzag) (voir la figure 6.2 (b)). Dans le cas du scénario de conduite normale, le modèle MRHLP ne segmente pas les données (voir figure 6.2 (f)) vu qu'on a un seul événement de conduite qui est la conduite normale. Contrairement au modèle MRHLP, le modèle HMM présente de nombreuses "pseudo" transitions (voir figure 6.2 (i)). Ce résultat s'explique par la flexibilité du processus logistique qui régit la transition d'un segment à un autre.

Les tableaux 6.1 et 6.2 donnent en détail les résultats de la segmentation des séquences de conduite obtenus par les deux approches. Comme on peut le voir sur le tableau 6.1, les temps de détection de chute et de presque chute sont quasiment les mêmes que les vrais instants. Ces résultats montrent la bonne qualité de segmentation des séquences de conduite, permettant ainsi un marquage précis sur les événements de conduite dans une base de données sans avoir recours à la visualisation des données vidéo. On note aussi que le modèle MRHLP permet d'avoir des taux de bonne segmentation avec 99% dans le cas des chutes, 95% dans le cas des presque chutes et de 96% dans le cas de conduite normale. Ce résultat est obtenu en comparant les résultats des segmentations aux véritables étiquettes; en évaluant toute correspondance possible. La correspondance fournissant le taux d'erreur de segmentation minimal est ainsi sélectionnée.

Dans le tableau 6.2, on observe les mêmes résultats que dans le cas du modèle MRHLP pour la détection du temps de chute. La détection des moments de chute avec

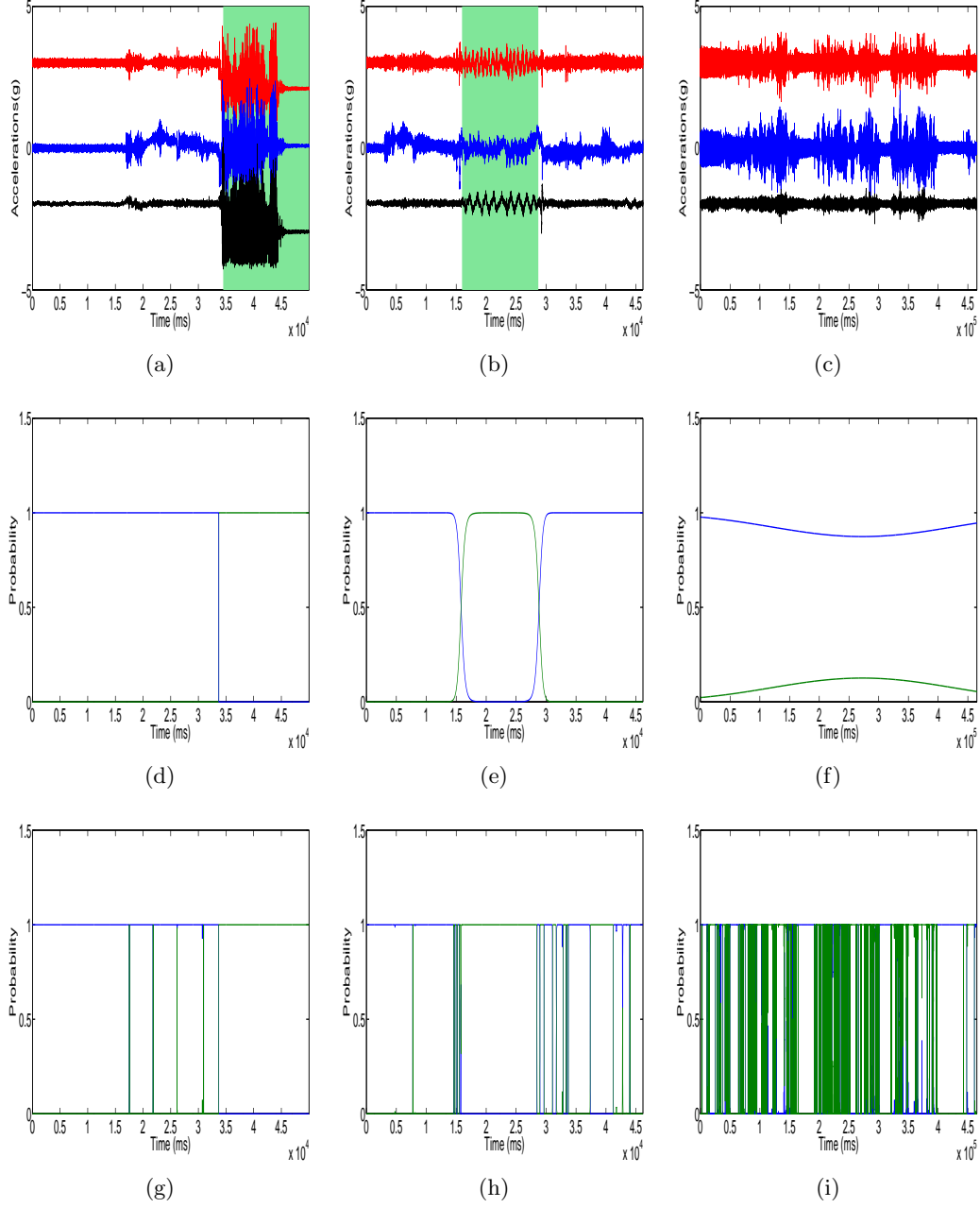


Figure 6.2: Résultats obtenus en appliquant le modèle MRHLP (au milieu) sur les signaux de la première ligne et le modèle HMM (en bas) sur trois séquences : chute, presque chute et normale (de gauche à droite).

6.4 Application de la méthodologie proposée et résultats

Table 6.1: Résultats de la segmentation avec le modèle MRHLP

Séquences	Temps d'occurrence de l'évènement (ms)	Temps de détection de l'évènement (ms)	Taux de bonne segmentation (%)
Séquence 1	14080	13810	99.10 %
Séquence 2	47070	46540	99.06%
Séquence 3	56380	55600	99.02%
Séquence 4	43750	42950	98.66%
Séquence 5	36160	35680	99.27%
Séquence 6	34540	33780	98.47%
Séquence 7	34500	33690	98.39%
Séquence 8	40420	38780	98.21%
Séquence 9	[14070, 24620]	[14110, 24670]	94.01%
Séquence 10	[13200, 26700]	[13190, 26740]	96.50%
Séquence 11	[14570, 24620]	[14670, 24670]	98.92%
Séquence 12	[867, 2422]	[1106, 3002]	96.34%
Séquence 13	[2355, 2526]	[2351, 2806]	95.34%
Séquence 14	[1104, 1757]	[1103, 1758]	98.42%
Séquence 15	[13930, 26830]	[14180, 26740]	98.14%
Séquence 16	[742, 1016]	[742, 1020]	87.65%
Séquence 17	[852, 1110]	[902, 1210]	90.38%
Séquence 18	[1597, 2871]	[1582, 2883]	98.08%
Séquences 19-29	0		95.72% ± 2

Table 6.2: Résultats de la segmentation avec le modèle HMM

Séquences	Temps d'occurrence de l'évènement (ms)	Temps de détection de l'évènement (ms)	Taux de bonne segmentation (%)
Séquence 1	14080	15700	94.83 %
Séquence 2	47070	47020	98.30 %
Séquence 3	56380	57670	71.04 %
Séquence 4	43750	42850	97.81 %
Séquence 5	36160	33700	95.32 %
Séquence 6	34540	33760	92.23 %
Séquence 7	34500	33700	97.99 %
Séquence 8	40420	37550	88.84 %
Séquence 9	[14070, 24620]	-	97.61%
Séquence 10	[13200, 26700]	-	96.14%
Séquence 11	[14570, 24620]	-	91.99%
Séquence 12	[867, 2422]	-	79.76%
Séquence 13	[2155, 2326]	-	87.38%
Séquence 14	[1104, 1757]	-	59.97%
Séquence 15	[13930, 26830]	-	95.63%
Séquence 16	[742, 1016]	-	86.07%
Séquence 17	[852, 1110]	-	78.0%
Séquence 18	[1597, 2871]	-	85.59%
Séquences 19-29	0		76.93% ± 7.23

le modèle HMM est relativement le même que les vrais instants de chute. On peut noter aussi que les taux de bonne segmentation obtenus environ 93% sont bons. Concernant les moments d'occurrence des événements de conduite de presque chute, ils ne sont pas mentionnés ici en raison de la présence de plusieurs "pseudo" transitions. Comme on peut le voir sur les taux de bonne segmentation (environ 86%) en moyenne, reflète la bonne efficacité du modèle HMM pour ce genre de problématique. Dans le cas de la conduite normale, le taux de bonne segmentation moyen sur les 11 séquences est de 77% qui est très bas par rapport à celui obtenu dans le cas du modèle MRHLP (96%).

Les résultats de segmentation en terme de taux de bonne segmentation de chaque algorithme et pour chaque type de séquences (chute, presque chute et normale) sont résumés dans le tableau 6.3. Nous remarquons que le modèle MRHLP effectue une meilleure segmentation dans tous les cas, comparé au modèle HMM.

Résultats de classification des événements critiques de conduite

Comme nous avons pu le constater dans les résultats présentés précédemment, l'étape de segmentation permet juste de savoir s'il y a présence d'un événement critique de

Table 6.3: Taux de bonne segmentation obtenu avec le modèle MRHLP et le modèle HMM.

	Séquence de chutes	Séquence presque chutes	Séquence de conduite normale	Taux global bonne segmentation
MRHLP (%)	98.77	95.37	95.72	96.62
HMM (%)	92.50	85.81	76.93	85.08

conduite ou pas et à quel moment. Et comme nous l'avons constaté aussi dans certains cas, il y avait présence de "pseudo" transitions et pour savoir si une transition est la conséquence de la présence d'un évènement critique de conduite (chute et presque chute) ou bien juste, une fausse détection, on a fait recours à un apprentissage supervisé. Pour cela, on s'intéresse aux paramètres estimés sur deux segments successifs. Dans cette étape les méthodes de segmentations présentées précédemment nous servent ici d'extracteur de caractéristiques. Ainsi le vecteur \mathcal{F} donné par :

$$\mathcal{F} = \{\beta_1, \beta_2, \|\beta_1 - \beta_2\|, \text{diag}(\Sigma_1), \text{diag}(\Sigma_2), \|\text{diag}(\Sigma_1 - \Sigma_2)\|\}$$

est utilisé comme base de données d'apprentissage pour classifier les différentes séquences de conduite en utilisant l'algorithme des k -ppv. Le but de cette étape est de juger si la transition entre deux segments contigus d'une séquence de conduite est une conséquence d'un évènement de conduite critique, qui peut être vu comme un changement important entre la moyenne et la variance d'un segment à un autre.

Pour aller plus en détail dans l'analyse des résultats obtenus, nous considérons l'ensemble des données constituées du vecteur \mathcal{F} de chaque séquence estimés par les approches de segmentation est divisé en un ensemble d'apprentissage et un ensemble de test selon une procédure de validation croisée 10-fois.

Les résultats de classification selon le vecteur \mathcal{F} obtenus à partir des modèles MRHLP et HMM sont résumés respectivement dans les matrices de confusion des tableaux: 6.4 et 6.5. Nous pouvons observer que les confusions se produisent surtout dans les séquences de presque chutes. Cette confusion est plus importante quand on utilise les paramètres \mathcal{F} issus du modèle HMM que ceux issus du modèle MRHLP avec respectivement 30% et 10%. Cette différence peut être expliquée par le fait que les caractéristiques extraites (\mathcal{F}) du modèle MRHLP sont plus discriminantes que celles extraites du modèle de HMM. Toutefois, les résultats dressés dans les tableaux 6.4 et 6.5 montrent qu'on arrive à bien classifier les séquences de conduite avec chute et les séquences de conduite normales avec un taux de 100% dans les deux cas de figure (RHLP et HMM).

Table 6.4: Matrice de confusion globale avec les paramètres issus du modèle MRHLP

		Classes obtenues		
		Chute	Presque chute	Conduite normale
Vraies classes	Chute (%)	100	0	0
	Presque chute (%)	0	90	10
	Conduite normale (%)	0	0	100

Table 6.5: Matrice de confusion globale avec les paramètres issus du modèle HMM

		Classes obtenues		
		Chute	Presque chute	Conduite normale
Vraies classes	Chute (%)	100	0	0
	Presque chute (%)	0	70	30
	Conduite normale (%)	0	0	100

6.4.2 Résultats de la détection de chute via la carte de contrôle MCUSUM

Dans cette section, nous présentons, analysons et comparons les résultats de la détection de chute obtenus par la méthode proposée à base de carte de contrôle et celle proposée par [74]. Nous considérons la base de données constituée par les 29 séquences de conduite présentées précédemment.

Mise en œuvre de la carte de contrôle MCUSUM

La détection d'anomalie se fait sur des fenêtres glissantes de taille W dont le choix dépend essentiellement de l'application visée. Dans le cas des problématiques de détection d'anomalies nécessitant une prise de décision rapide, il est recommandé d'utiliser des fenêtres de petite taille [95]. Dans notre cas d'étude, comme notre objectif est de détecter la chute le plus tôt possible afin d'avoir le temps pour déclencher le système de gonflage du gilet air-bag, la taille de la fenêtre W est fixée à 10 échantillons. Ce choix est lié d'une part à la fréquence d'acquisition des capteurs (1000hz) et d'autre part afin d'assurer l'hypothèse de normalité des observations sur des fenêtres locales. Ici on se place dans le cas de détection de chutes en ligne sous les deux hypothèses H_0 (conduite sans chute) et H_1 (présence d'une chute à l'instant t_k). La mise en œuvre d'un système de détection de chute de deux-roues motorisés en utilisant la carte de contrôle MCUSUM est divisée en deux phases comme suit:

- **Une phase d'apprentissage** (hors ligne) qui consiste à sélectionner les observations de la base de données dont le comportement est normal afin d'estimer la moyenne μ_0 et la matrice de covariance Σ_0 . Dans le cas des séquences de chute, la population sous contrôle (avant la chute) est sélectionnée en utilisant les

informations provenant de la caméra (détermination du moment de la chute).

- **Une phase de détection** (en ligne) qui consiste à appliquer les tests séquentiels sur de nouveaux individus en calculant et en comparant la statistique \mathbf{g}_{t_k} au seuil h afin de détecter la présence ou non de chute de la moto. Les cartes de contrôle MCUSUM donnent une sortie du signal de commande si la statistique calculée \mathbf{g}_{t_k} sur un individu à l'instant t_k est supérieure au seuil h .

Il est à noter que la procédure du *leave one out* est utilisée ici pour évaluer les performances de la carte MCUSUM. En effet, la phase d'apprentissage (estimation de la moyenne $\boldsymbol{\mu}_0$ et de la matrice de covariance $\boldsymbol{\Sigma}_0$) est effectuée sur 28 séquences et le test est effectué sur la séquence restante. Cette procédure est itérée 29 fois pour évaluer les performances de la carte de contrôle. Les critères d'évaluation des performances que nous avons utilisés sont la courbe ROC et le temps de détection des chutes.

Réglage des paramètres de la carte de contrôle MCUSUM

La limite de contrôle h de la carte MCUSUM ainsi que le paramètre k sont déterminés en utilisant le temps moyen entre deux fausses alarmes données par l'ARL (Average Run Length) [100].

$$ARL = \frac{1}{\alpha} \quad (6.27)$$

où α est la probabilité de fausses alarmes. Par exemple, lorsque le processus est sous contrôle avec $\alpha = 0.005$, alors l'ARL est égal à $\frac{1}{0.005} = 200$. Ce qui signifie que la carte de contrôle signale une fausse alarme (hors du contrôle) en moyenne toutes les 200 ms. Des limites de contrôle théoriques ainsi que le paramètre k associé au nombre de variables de données d'entrée ont été donnés par [100]. Dans [100], deux courbes d'évolution d'ARL sont données, à savoir l'ARL pour un taux de fausses alarmes espéré $\alpha = 0,2$ et l'ARL pour un taux de fausses alarmes espéré $\alpha = 0,5$ avec les limites de contrôle h et le paramètre k associés à chaque nombre de variables d'entrées sous l'hypothèse que la loi sous-jacente est normale.

Dans notre application, l'hypothèse de normalité n'est pas respectée par les données collectées, les seuils théoriques ne sont pas adaptés à la carte de contrôle. Par conséquent, nous avons estimé les nouveaux seuils h et le paramètre k en se fondant sur un compromis entre le taux de fausses alarmes et le taux de bonnes détection à partir de la courbe ROC.

Pour estimer le seuil h et le paramètre k de la carte de contrôle MCUSUM, nous avons adopté un $\alpha = 0,005$, c.à.d l'ARL est égal à $\frac{1}{0.005} = 200$.

L'estimation de ces paramètres est effectuée en deux étapes:

1. *La première étape* consiste à sélectionner le paramètre k . Pour cela, nous avons

fait varier le paramètre k de 3 à 12 et pour chaque valeur de k nous traçons la courbe ROC correspondante en faisant varier h de 0 à 1000 (voir la figure 6.3). En se fondant sur le résultat de l'aire sous la courbe (AUC) de chaque courbe ROC, nous avons sélectionné $k = 10$, qui correspond à la valeur la plus élevée de AUC (AUC=0.99).

2. La deuxième étape consiste à sélectionner la limite de contrôle h . Pour cela, on utilise la courbe ROC du k choisi en faisant varier h et on sélectionne la valeur de h qui conduit à un meilleur compromis entre la précision de détection de chute et le taux de fausses alarmes. Dans notre cas, nous avons utilisé la courbe ROC de $k = 10$ à partir de laquelle nous avons sélectionné la limite de contrôle $h = 70$ qui donne le meilleur compromis entre la précision de détection de chute et le taux de fausses alarmes.

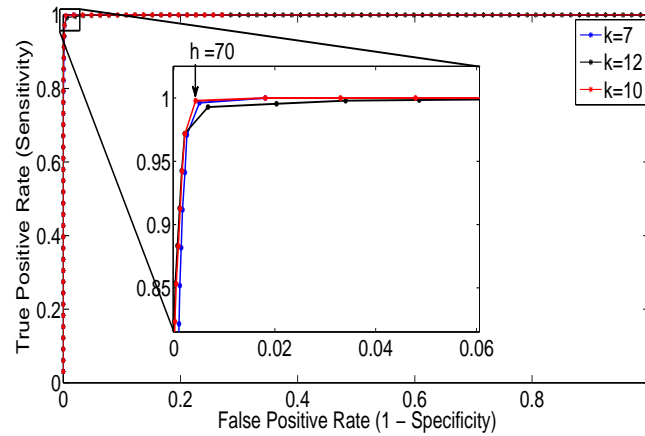


Figure 6.3: Exemple de courbes ROC obtenues pour $k=1$ (rouge), $k=2$ (bleu), $k=3$ (noir).

La figure 6.4 représente la statistique g_t calculée pour trois scénarios. On peut observer que dans le cas du scénario de chute (cf. la figure 6.4.(a)), la statistique g_t explose et dépasse le seuil h au moment où la moto commence à chuter. Dans le cas de presque chute, la statistique g_t calculée pour la séquence 12 est présentée sur la figure 6.4.(b). On note que dans cette séquence une accélération brusque a été effectuée par le conducteur. Cet événement peut être vu à travers la statistique calculée g_t par la présence d'un pic (sans dépasser le seuil h) au moment de cette accélération agressive. La figure 6.4.(c) représente la statistique g_t calculée dans le cas d'une séquence de conduite normale (séquence 22). Trois petits pics (sans dépasser le seuil h) peuvent être identifiés. Après l'analyse vidéo, nous avons remarqué que le premier pic est dû à la présence d'un nid de poule sur la route, comme illustré sur la figure 6.5, le deuxième pic est dû au passage de moto sur un ralentisseur routier et le dernier correspond à la

6.4 Application de la méthodologie proposée et résultats

présence de pavés sur la route.

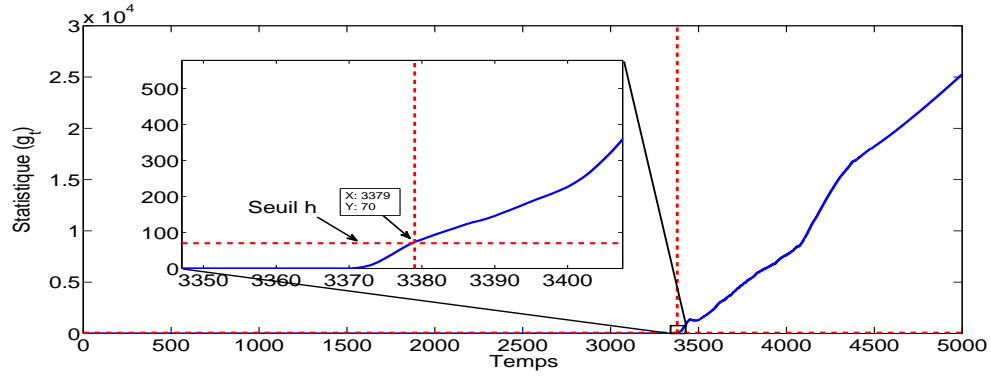
Comme indiqué ci-dessus, les deux modèles de détection de chute sont évalués sur une base de données réelle. Cette base de données est constituée de 29 séquences dans lesquelles nous avons 8 séquences de chute avec cinq scénarios, 10 séquences de presque chute représentant six scénarios et 11 séquences de conduite normale.

Les résultats de détection de chute obtenus sur toutes les séquences en terme de temps de détection de chutes utilisant la méthodologie proposée et la méthodologie proposée par [74] sont présentés dans le tableau 6.6.

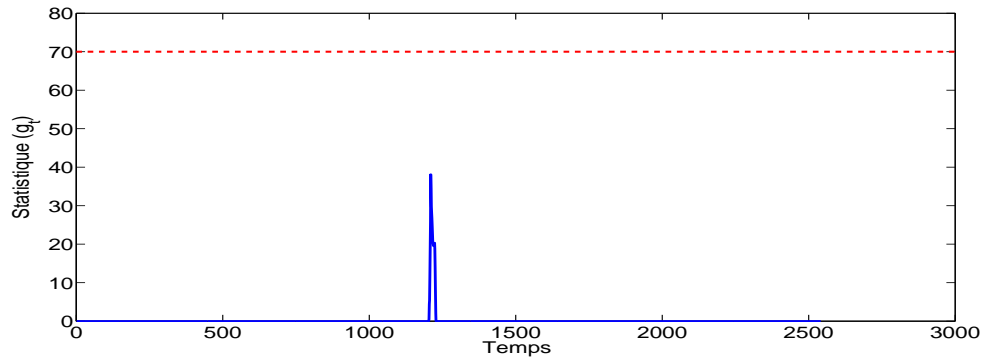
Table 6.6: Résultats obtenus en termes de temps de détection de chute avec la méthode proposée et la méthode de la norme.

Séquences	Moment de chute t (ms)	Moment de détection t_m (ms) MCUSUM	$t-t_m$	Moment de détection t_n (ms) [74]	$t-t_m$ [74]
Séquence 1	14080	13290	+790	14090	<u>-10</u>
Séquence 2	47070	46200	+870	47100	<u>-30</u>
Séquence 3	56380	55780	+600	56010	+370
Séquence 4	43750	43460	+290	43460	+290
Séquence 5	36160	35740	+420	35800	+360
Séquence 6	34540	34020	+520	34860	<u>-320</u>
Séquence 7	34500	33790	+710	34310	+190
Séquence 8	40420	39740	+680	40060	+360
Séquence 9	0	0	0	0	0
Séquence 10	0	0	0	0	0
Séquence 11	0	0	0	0	0
Séquence 12	0	0	0	0	0
Séquence 13	0	0	0	<u>22620</u>	<u>-22620</u>
Séquence 14	0	0	0	0	0
Séquence 15	0	0	0	0	0
Séquence 16	0	0	0	0	0
Séquence 17	0	0	0	0	0
Séquence 18	0	0	0	0	0
Séquences 19-29	0	0	0	0	0

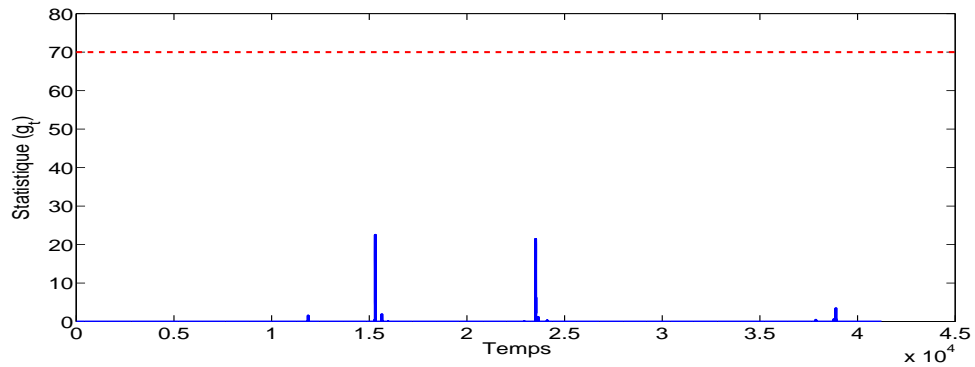
Nous pouvons remarquer que les résultats obtenus avec la méthode proposée sont nettement meilleurs que ceux obtenus par la méthode fondée sur le calcul de la norme, à la fois en termes de temps de détection et de taux de fausses alarmes. Nous pouvons aussi voir que dans le cas des séquences de chute, l'algorithme proposé détecte la chute avec un temps minimum de 290 ms, lequel est largement suffisant pour déclencher le gonflage du gilet air-bag porté par le conducteur (la durée typique de gonflage total d'un gilet air-bag étant de 100 ms). Quant à l'algorithme basé sur le calcul de la norme, on note dans le cas des séquences de chutes (séquences 1, 2 et 6), une détection tardive



(a)



(b)



(c)

Figure 6.4: La statistique g_t calculée pour : (a) scénario de chute, (b) scénario de presque chute et (c) conduite naturelle



Figure 6.5: Une capture d'écran de la vidéo enregistrée durant une séquence de conduite (séquence 22), au passage de la moto sur un nid de poule

de la chute avec des retards de détection respectifs de 10 ms, 30 ms et 320 ms. On note aussi que dans le cas des séquences de presque chute et de conduite normale, la carte de contrôle MCUSUM ne détecte aucune anomalie contrairement à l'algorithme basé sur le calcul de la norme. Ce dernier détecte une anomalie (synonyme de chute) dans la séquence 13.

6.5 Conclusion

Dans ce chapitre, nous avons présenté une méthodologie de détection et de classification des événements critiques dans la conduite. Le problème de détection des événements critiques dans la conduite d'un 2RM a été formulé comme celui de détection des changements des statistiques (moyenne et variance) d'un régime de conduite à un autre. La détermination des régimes ainsi que l'estimation des moyennes et des variations associées à chaque régime est réalisée par une segmentation jointe des séries temporelles multidimensionnelles. Deux approches ont été proposées à savoir le modèle de mélange de gaussiennes avec des proportions logistiques qui est un cas particulier du modèle RHLP et le modèle de Markov caché (HMM). De par leur formulation, les modèles HMM et MRHLP répondent mieux à la fois au problème de l'hétérogénéité des séries temporelles associées aux signaux accélérométriques et gyroscopiques et à l'aspect dynamique de ces séries. Ces méthodes servent en effet à la segmentation des signaux afin

de détecter des événements critiques dans l'épisode de conduite d'une part, et d'autre part à la détermination des plages temporelles d'occurrence de ces événements.

La classification des événements de conduite détectés est réalisée par l'algorithme k -ppv. Ce dernier prend comme entrées les paramètres des modèles estimés (les moyennes et les variances). Pour la validation de notre approche, nous avons utilisé les données recueillies à partir d'une moto instrumentée. Les résultats de segmentation obtenus sont très encourageants et montrent la possibilité de détecter non seulement un événement de conduite dans une base de données mais aussi la plage temporelle de son occurrence. Ces résultats montrent aussi la supériorité de l'approche fondée sur le modèle RHLPP par rapport à l'approche fondée sur le modèle HMM en termes de précision de la segmentation, aussi bien que dans la classification.

Le problème de détection de chute des 2RMs est formulé comme un problème de détection d'anomalie via une carte de contrôle MCUSUM. La carte MCUSUM a été appliquée sur la base de données réelles et a été comparée à l'algorithme proposé par [74]. Ces travaux nous ont permis de valider expérimentalement la méthode proposée pour la détection de chute des 2RMs. Les résultats obtenus montrent l'efficacité et la robustesse de la carte de contrôle MCUSUM pour résoudre ce type de problématique. La carte de contrôle MCUSUM permet de détecter les chutes avec un temps de détection minimum de 290 ms avant l'impact du conducteur sur terre dans le cas des séquences de chutes, ce qui est largement suffisant pour déclencher un système de gonflage du gilet air-bag. Le principal avantage de cette méthode réside dans le fait qu'elle est simple à implémenter en temps réel. Pour la mise en œuvre en temps réel de cette méthode, seule l'étape de détection sera considérée en ligne, l'étape d'apprentissage, se fera hors ligne.

Chapitre 7

Conclusion et Perspectives

Ces travaux de thèse avaient pour objectif principal la mise au point d'outils d'analyse automatique de signaux acquis sur un 2RM, accélérations et vitesses de rotation dans les trois dimensions (X-Y-Z), en se basant sur l'apprentissage statistique. Ces travaux sont motivés par un contexte général lié à l'augmentation du trafic des 2RMs, la vulnérabilité des usagers de ce moyen de transport et le manque d'études et de connaissances des pratiques réelles de conduite des 2RMs.

Après avoir décrit les différents capteurs installés sur le 2RM, les données recueillies et la base de données composée des séquences de conduite à la fois normale, à risque et avec chute, nous avons présenté les différents pré-traitements de données qui consistent principalement au remplacement des valeurs manquantes et aberrantes, au filtrage du bruit ainsi que la normalisation des données. Ces pré-traitements spécifiques sont effectués pour augmenter le pouvoir descriptif et discriminant des données de mesures.

La première problématique traitée dans cette thèse porte sur la reconnaissance automatique de situations de conduite à l'aide d'approches de classification proposée (chapitre 5). Pour ce faire, nous avons eu recours à une méthodologie modulaire simple, fondée sur des approches de classification supervisée à savoir: le modèle de mélange de gaussiennes, les k -plus proches voisins, les séparateurs à vaste marge, les forêts aléatoires et les modèles de Markov cachés. Les résultats obtenus montrent l'efficacité de l'approche des HMMs dans les deux cas continu et discret. Ce résultat est attendu de fait que les HMMs sont des modèles qui tiennent compte de l'aspect temporel des données ce qui est le cas des signaux traités. Afin d'améliorer les résultats obtenus, nous avons proposé pour ce type d'application une procédure de sélection de capteurs à l'aide CHMMs qui a permis d'améliorer les résultats de la classification des situations de conduite.

La seconde problématique abordée dans ces travaux de thèse porte sur la détection d'événements de conduite à risque (chapitre 6). La première partie est consacrée au problème de détection de chute des 2RMs dans la perspective d'élaborer un système

de déclenchement de gonflage du gilet air-bag porté par le conducteur de 2RM. Pour résoudre ce problème, nous avons proposé une nouvelle méthode de détection de chute des 2RMs fondée sur les cartes de contrôle MCUSUM. Les résultats obtenus sur une base de données constituée de séquences de conduite avec chute, séquences de conduite avec presque chute et des séquences de conduite normale montrent l'efficacité et la robustesse de l'approche proposée. La deuxième partie est consacrée au développement et à la mise en œuvre de méthodes de détection d'événements critiques de conduite dans un flux de données, cela dans l'objectif de faciliter l'analyse et la compréhension des événements critiques de conduite. Cette analyse servira non seulement à la proposition de contre-mesures afin d'assister le conducteur mais aussi à améliorer la qualité de formation des conducteurs. Dans ce contexte, nous avons formulé cette question complexe comme un problème de détection de changement de statistiques (moyennes et variances) des signaux acquis, d'un régime de conduite à un autre au cours d'une séquence. Le problème de la détection des événements critiques a été réalisé en deux étapes: (1) **une étape de segmentation** et d'extraction de caractéristiques (moyennes et variances) où les séquences de conduite ont été modélisées et segmentées en utilisant un modèle de régression régi par un processus logistique caché désigné par MRHLP (Multiple Regression Hidden Logistic Process); (2) **une étape de classification** qui consiste à utiliser l'algorithme des k -plus proches voisins (k-ppv) pour affecter chaque segment à l'une des trois classes à savoir conduite normale, conduite avec presque accident et conduite avec chute, selon les caractéristiques extraites via les paramètres du modèle MRHLP. Les résultats obtenus montrent la capacité de la méthodologie proposée à détecter des événements de conduite critiques des 2RMs.

Ces travaux peuvent être étendus à différents niveaux. Tout d'abord, dans le cadre de la classification de situations de conduite les résultats de la classification sont obtenus en utilisant les données issues d'une centrale inertielle (accéléromètre/gyroscope). Ces résultats peuvent être améliorés en incluant une étape d'extraction de caractéristiques sur les données brutes. En outre, les méthodes d'apprentissage supervisé sont utilisées dans ce travail. Ces méthodes exigent l'annotation (la labélisation) des données pour effectuer l'apprentissage des modèles. La tâche d'annotation manuelle peut introduire des erreurs d'étiquetage, en particulier dans les phases de transitions entre les différentes situations de conduite. Plusieurs solutions peuvent être envisagées, soit en prenant en compte dans l'apprentissage des modèles, les erreurs possibles d'étiquetage [111], [112]; soit en utilisant l'apprentissage non supervisé adapté aux données séquentielles [113]. La fusion des sorties de plusieurs classifieurs notamment dans le cadre de la théorie des fonctions de croyance (Dempster-Shafer) [114] en vue de limiter l'impact d'éventuels conflits entre sous-classifieurs peut aussi être envisagée comme une solution robuste à ce problème [115].

Une autre perspective de ce travail est d'utiliser la notion de conflit entre les dif-

férentes sources de données (capteurs installés sur la moto) comme un indicateur de l'occurrence d'évènements «chute ou presque chute» dans l'épisode de conduite. Il s'agit alors de mettre au point des indicateurs de comportement ou de situations de conduite basés sur la théorie des fonctions de croyance. Ce cadre crédibiliste, permettra en effet de mieux représenter des informations imparfaites pouvant être aussi bien incertaines qu'imprécises, ce qui n'est pas le cas dans le cadre de la théorie classique des probabilités.

Bibliographie

- [1] V. Cossalter, *Motorcycle dynamics*. Race Dynamics, Greendale, WI, 2002, Lulu.com, 2006.
- [2] B. A. Jonah, “Sensation seeking and risky driving: a review and synthesis of the literature,” *Accident Analysis and Prevention*, vol. 29, pp. 651–665, 1997.
- [3] M. S. Foundation, “100 motorcyclists naturalistic study, <http://www.motorcycle-usa.com/2/14113/motorcycle-article/msf-100-motorcyclists-naturalistic-study.aspx>.”
- [4] D. Mitrovic, “Learning driving patterns to support navigation,” Ph.D. dissertation, Phd Thesis, University of Canterbury, 2004.
- [5] B. Larnaudie, “Codesign, architecture fonctionnelle de fusion et architecture capteurs pour l’identification de situations accidentogènes: application à la sécurisation de véhicules deux-roues,” Ph.D. dissertation, Doctorat de l’ Université Paris 11, 2006.
- [6] N. Dapzol, “Analyse de l’activité de conduite par les chaînes de markov cachées et les modèles de ruptures multi-phasiques: méthodologie et applications,” Ph.D. dissertation, Doctorat de l’Université Claude Bernard-Lyon I, 2006.
- [7] S. Aupetit, J. Riff, V. Gallier, and S. Espié, “Conceptual and methodological tools for analyzing situations of vulnerability on the road: examples from a motorcycle riding study,” in *Proceedings of the 30th European Conference on Cognitive Ergonomics*, 2012, pp. 29–36.
- [8] W.-z. Wang, Y.-W. Guo, B.-Y. Huang, G.-R. Zhao, B.-q. Liu, and L. Wang, “Analysis of filtering methods for 3d acceleration signals in body sensor network,” in *Bioelectronics and Bioinformatics (ISBB), 2011 International Symposium on*. IEEE, 2011, pp. 263–266.
- [9] T. Mitchell, *Machine Learning*. McGraw Hill, New York, 1997.
- [10] S.-B. Cho and H.-H. Won, “Machine learning in dna microarray analysis for cancer classification,” in *Proceedings of the First Asia-Pacific bioinformatics conference*

- on Bioinformatics 2003-Volume 19.* Australian Computer Society, Inc., 2003, pp. 189–198.
- [11] A. Lapedes, C. Barnes, C. Burks, R. Farber, and K. Sirotkin, “Application of neural networks and other machine learning algorithms to dna sequence analysis,” Bell.G. and Marr.T.(eds). *Computers and DNA*. Addison-Wesley, Redwood City, CA, Tech. Rep., 1995.
- [12] N. Bhardwaj, R. E. Langlois, G. Zhao, and H. Lu, “Kernel-based machine learning protocol for predicting dna-binding proteins,” *Nucleic Acids Research*, vol. 33, no. 20, pp. 6486–6493, 2005.
- [13] A. F. Atiya, “Bankruptcy prediction for credit risk using neural networks: A survey and new results,” *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 929–935, 2001.
- [14] G. Zhang, M. Y. Hu, B. E. Patuwo, and D. C. Indro, “Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis,” *European journal of operational research*, vol. 116, no. 1, pp. 16–32, 1999.
- [15] K. Torkkola, S. Venkatesan, and H. Liu, “Sensor selection for maneuver classification,” in *Proceedings of the 7th IEEE International ITSC Conference*, 2004, pp. 636–641.
- [16] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 4, no. 4.
- [17] P. Dayan, “Unsupervised learning,” *Wilson, R. A. and Keil, F., editors. The MIT Encyclopedia of the Cognitive Sciences*, 1999.
- [18] J. H. Ward Jr, “Hierarchical grouping to optimize an objective function,” *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.
- [19] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. California, USA, 1967, pp. 281–297.
- [20] M. Geoffrey and D. Peel, *Finite mixture models*. John Wiley & Sons, 2004.
- [21] J. A. Lee and M. Verleysen, *Nonlinear dimensionality reduction*. New York; London: Springer, 2007.
- [22] H. Liu, H. Motoda, R. Setiono, and Z. Zhao, “Feature selection: An ever evolving frontier in data mining.” in *The Fourth Workshop on Feature Selection in Data Mining*, 2010, pp. 4–13.

- [23] T. Marill and D. M. Green, "On the effectiveness of receptors in recognition systems," *IEEE Transactions on Information Theory*, vol. 9, no. 1, pp. 11–17, 1963.
- [24] A. W. Whitney, "A direct method of nonparametric measurement selection," *IEEE Transactions on Computers*, vol. 100, no. 9, pp. 1100–1103, 1971.
- [25] J. Kittler, "Feature set search algorithms," *Pattern recognition and signal processing*, pp. 41–60, 1978.
- [26] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern recognition letters*, vol. 15, no. 11, pp. 1119–1125, 1994.
- [27] H. Almuallim and T. G. Dietterich, "Learning with many irrelevant features." in *Proceedings of the Ninth National Conference on Artificial Intelligence*, vol. 91. MIT Press, 1991, pp. 547–552.
- [28] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 1992, pp. 129–134.
- [29] —, "A practical approach to feature selection," in *Proceedings of the ninth international workshop on Machine learning*. Morgan Kaufmann Publishers Inc., 1992, pp. 249–256.
- [30] S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection," in *Proceedings of the International Conference on Machine Learning*, vol. 1. Cite-seer, 2001, pp. 74–81.
- [31] L. Ljung, *System Identification - Theory for User*. Prentice-Hall ed. NJ, 1987.
- [32] M. Stone, "Cross validation choice and assessment of statistical predictions," *Journal of the Royal Statistical Society*, vol. B(36), pp. 111–147, 1974.
- [33] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *IJCAI*, pp. 1137–1145, 1995.
- [34] B. Efron and R. J. Tibshirani, *An introduction to the Bootstrap*, ser. Monographs on Statistics and Applied Probability. Chapman and Hall, 1998, vol. 57.
- [35] B. Efron, "Estimating the error rate of a prediction rule; improvement on cross-validation," *Journal of the American Statistical Association*, vol. 78, pp. 316–331, 1983.
- [36] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 16, no. 4, pp. 37–46, 1960.

-
- [37] L. J.R and K. G.G, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 59–174, 1977.
- [38] A. F. Clark and C. Clark, "Performance characterization in computer vision a tutorial," 1999.
- [39] M. C. Nechyba and Y. Xu, "Stochastic similarity for validating human control strategy models," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 3, pp. 437–451, 1998.
- [40] A. Pentland and A. Liu, "Modeling and prediction of human behavior," *Neural Comput.*, vol. 11, no. 1, pp. 229–242, 1999.
- [41] K. Torkkola, S. Venkatesan, and H. Liu, "Sensor sequence modeling for driving," in *FLAIRS Conference*, 2005, pp. 721–727.
- [42] D. Mitrovic, "Reliable method for driving events recognition," *Trans. Intell. Transport. Sys.*, vol. 6, no. 2, pp. 198–205, 2005.
- [43] N. Thepvilojanapong, K. Sugo, Y. Namiki, and Y. Tobe, "Recognizing bicycling states with hmm based on accelerometer and magnetometer data," in *Proceedings of IEEE on SICE Annual Conference (SICE)*, 2011, pp. 831–832.
- [44] D. M. Titterington, A. F. Smith, U. E. Makov *et al.*, *Statistical analysis of finite mixture distributions*. Wiley New York, 1985, vol. 7.
- [45] G. McLachlan and K. Basford, "Mixture models: inference and applications to clustering. 1988," *Marcel and Dekker Inc., New York*, 1988.
- [46] M. Stephens and D. Phil, "Bayesian methods for mixtures of normal distributions," 1997.
- [47] M. Stephens, "Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods," *Annals of Statistics*, pp. 40–74, 2000.
- [48] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.
- [49] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. Wiley, New York, 1997.
- [50] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

- [51] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, 2001.
- [52] N. Bhatia *et al.*, “Survey of nearest neighbor techniques,” *arXiv preprint arXiv:1007.0085*, 2010.
- [53] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
- [54] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.
- [55] L. Breiman, “Bagging predictors,” *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [56] —, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [57] Y. Amit, D. Geman, and K. Wilder, “Joint induction of shape features and tree classifiers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 11, pp. 1300–1305, 1997.
- [58] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state markov chains,” *The Annals of Mathematical Statistics*, vol. 37, pp. 1554–1563, 1966.
- [59] Y. Bengio and P. Frasconi, “An input output hmm architecture,” *Advances in neural information processing systems*, pp. 427–434, 1995.
- [60] G. Celeux, J. Nascimento, and J. Marques, “Learning switching dynamic models for objects tracking,” *Pattern Recognition*, vol. 37, no. 9, pp. 1841–1853, 2004.
- [61] S. Frühwirth-Schnatter, *Finite Mixture and Markov Switching Models: Modeling and Applications to Random Processes*. Springer, 2006.
- [62] B.-H. Juang and L. Rabiner, “Mixture autoregressive hidden markov models for speech signals,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, no. 6, pp. 1404–1413, 1985.
- [63] F. Muri, “Comparaison d’algorithmes d’identification de chaînes de markov cachées et application à la détection de régions homogènes dans les séquences d’adn,” Ph.D. dissertation, Université Paris V, 1997.
- [64] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” in *Proceedings of the IEEE*, 1989, pp. 257–286.
- [65] K. P. Murphy, “Hidden semi-markov models (hsmms),” unpublished notes, Tech. Rep., 2002.

-
- [66] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.
- [67] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *The annals of mathematical statistics*, pp. 164–171, 1970.
- [68] S. Preece, J. Goulermas, L. Kenney, and D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *IEEE Transactions on Biomedical Engineering*, vol. 56, pp. 871–879, 2009.
- [69] E. I. Vlahogianni, G. Yannis, and J. C. Golias, "Critical power two wheeler driving patterns at the emergence of an incident," *Accident Analysis & Prevention*, vol. 58, pp. 340–345, 2013.
- [70] F. Chamroukhi, "Hidden process regression for curve modeling, classification and tracking," Ph.D. Thesis, Université de Technologie de Compiègne, Compiègne, France, 2010.
- [71] S. Iijima, "Exploratory study of an airbag concept for a large touring motorcycle," in *16th International Technical Conference on the Enhanced Safety of Vehicles*, 1998, 1998.
- [72] T. Kuroe, N. Namiki, and S. Iijima, "Exploratory study of an airbag concept for a large touring motorcycle: Further research second report," in *19th International Technical Conference on the Enhanced Safety Vehicles*, 2005.
- [73] T. Yamazaki, S. Iijima, and T. Yamamoto, "Exploratory study of an airbag concept for a large touring motorcycle: Further research," in *17th International Technical Conference on the Enhanced Safety of Vehicles*, 2001.
- [74] A. Boubezoul, S. Espié, B. Larnaudie, and S. Bouaziz, "A simple fall detection algorithm for powered two wheelers," *Control Engineering Practice*, pp. 286–297, 2013.
- [75] V. Cossalter, A. Bellati, and V. Cafaggi, "Exploratory study of the dynamic behaviour of motorcycle-rider during incipient fall events," in *In the 19th International Technical Conference on the Enhanced Safety of Vehicles Conference (ESV) in Washington, D.C.*, 6-9 June 2005.
- [76] V. Cossalter, A. Aguggiaro, D. Debus, A. Bellati, and A. Ambroggi, "Real cases motorcycle and rider race data investigation: Fall behavior analysis," in *In the 20th Enhanced Safety of Vehicles Conference, Innovations for Safety Opportunities and Challenges, Lyon, France*, june 2009.

-
- [77] A. Bellati, V. Cossalter, R. Lot, and A. Ambrogi, "Preliminary investigation on the dynamics of motorcycle fall behavior: influence of a simple airbag jacket system on rider safety," in *In the 6th International Motorcycle Conference, Cologne, Germany*, 9-10 October 2006, pp. 05–0266.
- [78] B. Larnaudie, S. Bouaziz, T. Maurin, S. Espié, and R. Reynaud, "Experimental study of a motorcycle platform for dynamic model extraction," in *Proceedings of the IEEE International Conference on Information & Communication Technologies: from Theory to Applications, Damascus, Syria*, vol. 01, 2006, pp. 760–766.
- [79] E. Aleskerov, B. Freisleben, and B. Rao, "Cardwatch: A neural network based database mining system for credit card fraud detection," in *Proceedings of the Computational Intelligence for Financial Engineering (CIFEr)*, 1997, pp. 220–226.
- [80] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," *Artificial Intelligence Review*, 2005.
- [81] Y. Kou, C.-T. Lu, S. Sirwongwattana, and Y.-P. Huang, "Survey of fraud detection techniques," in *IEEE international conference on Networking, sensing and control*, vol. 2, 2004, pp. 749–754.
- [82] K. B. Bignell, "Authentication in an internet banking environment: towards developing a strategy for fraud detection," In: *Proc. IEEE of International Conference on Internet Surveillance and Protection (ICISP)*, Cote d'Azur, France, p. 23–30, 2006.
- [83] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *Journal of computer security*, vol. 6, no. 3, pp. 151–180, 1998.
- [84] D. M. Tax, "One-class classification," 2001.
- [85] M. Markou and S. Singh, "Novelty detection: a review—part 2:: neural network based approaches," *Signal processing*, vol. 83, no. 12, pp. 2499–2521, 2003.
- [86] —, "Novelty detection: a review—part 1: statistical approaches," *Signal processing*, vol. 83, no. 12, pp. 2481–2497, 2003.
- [87] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.
- [88] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection," in *Applications of data mining in computer security*. Springer, 2002, pp. 77–101.

- [89] L. Ertoz, E. Eilertson, A. Lazarevic, P.-N. Tan, V. Kumar, J. Srivastava, and P. Dokas, “Minds-minnesota intrusion detection system,” *Next Generation Data Mining*, pp. 199–218, 2004.
- [90] V. Chandola, E. Eilertson, L. Ertoz, G. Simon, and V. Kumar, “Data mining for cyber security,” *Data Warehousing and Data Mining Techniques for Computer Security*, pp. 83–103, 2006.
- [91] T. M. Luong, Y. Rozenholc, and G. Nuel, “Fast estimation of posterior probabilities in change-point analysis through a constrained hidden markov model,” *Computational Statistics & Data Analysis*, vol. 68, pp. 129–140, 2013.
- [92] C. Baysse, D. Bihannic, A. Gégout-Petit, M. Prenat, and J. Saracco, “Hidden markov model for the detection of a degraded operating mode of optronic equipment,” *arXiv preprint arXiv:1212.2358*, 2012.
- [93] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.
- [94] W. H. Woodall, “Controversies and contradictions in statistical process control,” *Journal of Quality Technology*, vol. 32, no. 4, pp. 341–350, 2000.
- [95] N. Cheifetz, A. Same, P. Akinin, and E. de Verdalle, “A pattern recognition approach for anomaly detection on buses brake system,” in *Proceeding of the 14th IEEE International Conference Intelligent Transportation Systems (ITSC)*, 2011, pp. 266–271.
- [96] A. Y. Cheng, R. Y. Liu, and J. T. Luxhøj, “Monitoring multivariate aviation safety data by data depth: control charts and threshold systems,” *IIE Transactions*, vol. 32, no. 9, pp. 861–872, 2000.
- [97] L. Jaupi, “Multivariate control charts for complex processes,” in *Multivariate Total Quality Control*. Springer, 2002, pp. 125–136.
- [98] S. T. A. Niaki and B. Abbasi, “Fault diagnosis in multivariate control charts using artificial neural networks,” *Quality and reliability engineering international*, vol. 21, no. 8, pp. 825–840, 2005.
- [99] R. Scranton, G. C. Runger, J. B. Keats, and D. C. Montgomery, “Efficient shift detection using multivariate exponentially-weighted moving average control charts and principal components,” *Quality and reliability engineering international*, vol. 12, no. 3, pp. 165–171, 1996.
- [100] R. B. Crosier, “Multivariate generalizations of cumulative sum quality-control schemes,” *Technometrics*, vol. 30, pp. 291–303, Aug. 1988.

- [101] E. Page, “Continuous inspection schemes,” *Biometrika*, pp. 100–115, 1954.
- [102] W. D. Ewan, “When and how to use cu-sum charts,” *Technometrics*, vol. 5, no. 1, pp. 1–22, 1963.
- [103] E. Page, “Cumulative sum charts,” *Technometrics*, vol. 3, no. 1, pp. 1–9, 1961.
- [104] —, “Controlling the standard deviation by cusums and warning lines,” *Technometrics*, vol. 5, no. 3, pp. 307–315, 1963.
- [105] F. Gan, “An optimal design of cusum quality control charts,” *Journal of Quality Technology*, vol. 23, no. 4, pp. 279–286, 1991.
- [106] J. M. Lucas, “The design and use of v-mask control schemes,” *Journal of Quality Technology*, vol. 8, no. 1, pp. 1–12, 1976.
- [107] —, “Combined shewhart-cusum quality control schemes,” *Journal of Quality Technology*, vol. 14, no. 2, pp. 51–59, 1982.
- [108] D. M. Hawkins, “Cumulative sum control charting: an underutilized spc tool,” *Quality Engineering*, vol. 5, no. 3, pp. 463–477, 1993.
- [109] —, “A cusum for a scale parameter,” 1979.
- [110] W. H. Woodall and B. M. Adams, “The statistical design of cusum charts,” *Quality Engineering*, vol. 5, no. 4, pp. 559–570, 1993.
- [111] B. Frénay, G. de Lannoy, and M. Verleysen, “Label noise-tolerant hidden markov models for segmentation: application to ecgs,” in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 455–470.
- [112] B. Frénay and M. Verleysen, “Classification in the presence of label noise: A survey,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 25, no. 5, pp. 845–869, 2014.
- [113] M. Fridman, “Hidden markov model regression,” Minneapolis, MN, Tech. Rep., 1993.
- [114] A. Roquel, “Exploitation du conflit entre capteurs pour la gestion d’un système complexe multi-capteurs,” Ph.D. dissertation, Doctorat de l’ Université Paris 11, 2012.
- [115] L. Xu, A. Krzyzak, and C. Y. Suen, “Methods of combining multiple classifiers and their applications to handwriting recognition,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 3, pp. 418–435, May 1992.

Table des figures

2.1	Évolution de la mortalité routière en France métropolitaine et les mesures prises en matière de sécurité routière entre 1970 et 2013: source ONISR.	6
2.2	Nombre de victimes par catégorie d’usagers de la route: source ONISR.	7
2.3	Types d’accidents par catégorie d’usagers de la route: source ONISR.	8
2.4	Organigramme de l’approche proposée dans ces travaux de thèse.	12
3.1	Moto instrumentée (Honda CBF1000) et capteurs installés. Les accélérations et les vitesses angulaires sont présentées dans le trièdre de référence.	15
3.2	Trajet réalisé par les cinq participants en conduite normale.	19
3.3	Piste d’essai CAROL, où les expériences de conduite à risque ont été réalisées.	20
3.4	Piste d’essai CERAM, où les expérimentations de conduite avec chute ont été réalisées.	22
3.5	Les vidéos enregistrées avec la camera rapide (1000 images par seconde) au cours d’une séquence de chute.	23
3.6	La densité spectrale de puissance de l’accélération longitudinale (ax) à gauche et son spectrogramme à droite de la figure.	25
3.7	Représentation des signaux accélérométriques et gyroscopiques bruts (en bleu) et filtrés (en rouge) enregistrés au cours d’une séquence de conduite normale.	27
3.8	Annotation des données en utilisant le logiciel ”BinAscii”.	28
3.9	L’accélération longitudinale enregistrée au cours d’une séquence de conduite normale et les classes correspondantes.	29
3.10	Représentation des proportions des classes de situation de conduite de chaque séquence de conduite normale (en haut) et pour toutes les séquences (en bas).	31
4.1	Processus de sélection des caractéristiques.	39

4.2	Approche Filtre.	40
4.3	Approche wrapper et hybride.	41
4.4	Exemples de modèle de courbes <i>ROC</i>	46
5.1	Synopsis de la méthodologie proposée.	54
5.2	Illustration de l'impact de la maximisation de la marge sur la capacité de généralisation. Le point gris est un élément de l'ensemble de test. . .	60
5.3	Illustration du modèle HMM gauche-droite.	67
5.4	Modèle graphique de génération des données d'un modèle HMM continu. .	68
6.1	Modèle graphique de génération de données d'un modèle MRHLP. . . .	84
6.2	Résultats obtenus en appliquant le modèle MRHLP (au milieu) sur les signaux de la première ligne et le modèle HMM (en bas) sur trois séquences : chute, presque chute et normale (de gauche à droite).	95
6.3	Exemple de courbes ROC obtenues pour $k=1$ (rouge), $k=2$ (bleu), $k=3$ (noir).	101
6.4	La statistique g_t calculée pour : (a) scénario de chute, (b) scénario de presque chute et (c) conduite naturelle	103
6.5	Une capture d'écran de la vidéo enregistrée durant une séquence de conduite (séquence 22), au passage de la moto sur un nid de poule	104

Liste des tableaux

3.1	Comparaison des valeurs de RSB et de R des trois filtres utilisés sur les signaux d'accélération et de vitesses angulaires.	27
3.2	Récapitulatif des différentes séquences de conduite réalisées	30
4.1	Matrice de Confusion dans le cas d'une classification binaire.	44
4.2	Interprétation des valeurs de κ_c	47
4.3	Table de contingence	47
5.1	Le taux de bonne classification des différents algorithmes d'apprentissage pour chaque séquence (le meilleur résultat est marqué en gras).	72
5.2	Les résultats obtenus en terme de F_1 -mesure par classe, F_1 -mesure global, κ_c et le taux de bonne classification global (R) et son écart type(Std) pour chaque algorithme (le meilleur résultat est marqué en gras).	73
5.3	Comparaison statistique des différents classifieurs via le test de McNemar Z. Les flèches (\leftarrow , \uparrow) indiquent l'algorithme le plus performant.	73
5.4	Le taux de bonne classification obtenu pour les différents sous-ensembles sélectionnés en utilisant le modèle CHMM	75
5.5	Le taux de bonne classification obtenu en utilisant le sous-ensemble SE_8	76
5.6	Les résultats obtenus en terme de F_1 -mesure par classe, F_1 -mesure global et le taux de bonne classification global (R) et son écart type (Std) pour les CHMMs en utilisant le sous-ensemble SE_8	76
5.7	Matrice de confusion globale en utilisant les six capteurs.	77
5.8	Matrice de confusion globale en utilisant le sous-ensemble SE_8	77
5.9	Le taux de bonne classification obtenu par les CHMMs en utilisant le sous-ensemble SE_8 (leave one out subject)	78
6.1	Résultats de la segmentation avec le modèle MRHLP	96
6.2	Résultats de la segmentation avec le modèle HMM	97

6.3	Taux de bonne segmentation obtenu avec le modèle MRHLP et le modèle HMM.	98
6.4	Matrice de confusion globale avec les paramètres issus du modèle MRHLP	99
6.5	Matrice de confusion globale avec les paramètres issus du modèle HMM	99
6.6	Résultats obtenus en termes de temps de détection de chute avec la méthode proposée et la méthode de la norme.	102

Liste de publications

Revues avec comité de lecture

- Attal, F., Boubezoul, A., Oukhellou, L., Espie, S. "Powered Two-Wheeler Riding Pattern Recognition Using a Machine-Learning Framework". Journal on Transactions Intelligent Transportation Systems, IEEE. January 2015.

Conférences internationales avec actes

- Attal, F., Boubezoul, A., Samé, A., Oukhellou, L. "Powered-Two-Wheeler safety critical events recognition using a mixture model with quadratic logistic functions". In 13th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges, Belgique, April, 2015.
- Attal, F., Boubezoul, A., Oukhellou, L., Cheifetz, N., Espié, S. "The Powered Two Wheelers fall detection using Multivariate CUMulative SUM (MCUSUM) control charts". In ITSC 17th International IEEE Conference On Transportation Systems. Qingdao, Chine, October, 2014.
- Attal, F., Boubezoul, A., Oukhellou, L., Espié, S. "Analysis and classification of Powered Two Wheelers riding pattern". In Transport Research Arena (TRA) 5th Conference: Transport Solutions from Research to Deployment. Paris, France, April, 2014.
- Attal, F., Boubezoul, A., Oukhellou, L., Espié, S. "Riding patterns recognition for Powered two-wheelers users' behaviors analysis". In ITSC 16th International IEEE Conference On Transportation Systems. The Hague, October, 2013.

Title: Powered Two Wheelers riding patterns classification and critical events recognition.

Abstract : This thesis aims to develop framework tools for analyzing and understanding the riding of Powered Two Wheelers (PTW). Experiments are conducted using instrumented PTW in real context including both normal (naturalistic) riding behaviors and critical riding behaviors (near fall and fall). The two objectives of this thesis are the riding patterns classification and critical riding events detection. In the first part of this thesis, a machine-learning framework is used for riding pattern recognition problem. Therefore, this problem is formulated as a classification task to identify the class of riding patterns. The approaches developed in this context have shown the interest to take into account the temporal aspect of the data in PTW riding. Moreover, we have shown the effectiveness of hidden Markov models for such problem. The second part of this thesis focuses on the development of the off-line detection and classification of critical riding events tools and the on-line fall detection. The problem of detection and classification of critical riding events has been performed towards two steps: (1) the segmentation step, where the multidimensional time of data were modeled and segmented by using a mixture model with quadratic logistic proportions; (2) the classification step, which consists in using a pattern recognition algorithm in order to assign each event by its extracted features to one of the three classes namely Fall, near Fall and Naturalistic riding. Regarding the fall detection problem, it is formulated as a sequential anomaly detection problem. The Multivariate CUMulative SUM (MCUSUM) control chart was applied on the data collected from sensors mounted on the motorcycle. The obtained results on a real database have shown the effectiveness of the proposed methodology for both riding pattern recognition and critical riding events detection problems.

Keywords : Powered Two Wheelers (PTW), riding pattern recognition, supervised and unsupervised machine learning, offline critical riding event detection, on line fall detection, anomalies detection.